NY property data explore

January 13, 2018

```
In [1]: import pandas as pd
        import numpy as np
        import scipy.stats as sps
        import matplotlib.pyplot as plt
        import seaborn as sns
        import sklearn as skl
        from sklearn import preprocessing
        %matplotlib inline

In [2]: %%time
        fa_dir = '/Users/stevecoggeshall/Documents/Teaching/Fraud Analytics'
        mydata = pd.read_excel(fa_dir + '/2018 USC fraud class/data/NY property/NY

CPU times: user 7min 20s, sys: 8.27 s, total: 7min 29s
Wall time: 7min 55s


In [3]: numrecords = len(mydata)
        print(numrecords)

1048575


In [4]: mydata.dtypes

Out[4]: RECORD          int64
        BBLE           object
        BLOCK           int64
        LOT             int64
        EASEMENT       object
        OWNER          object
        BLDGCL         object
        TAXCLASS       object
        LTFRONT         int64
        LTDEPTH         int64
        STORIES       float64
        FULLVAL         int64
        AVLAND          int64
        AVTOT         int64
        EXLAND        int64
        EXTOT         int64
        EXCD1       float64
        STADDR       object
        ZIP         float64
        EXMPTCL      object
        BLDFRONT      int64
        BLDDEPTH      int64
        AVLAND2     float64
        AVTOT2      float64
        EXLAND2     float64
        EXTOT2      float64
        EXCD2       float64
        PERIOD       object
        YEAR         object
        VALTYPE      object
        dtype: object
```

```
In [5]: mydata.head(10).transpose()
```

```
Out[5]:                            0                   1                   2  \
        RECORD                     1                   2                   3
        BBLE              3046020035          5046820019          3074790028
        BLOCK                   4602                4682                7479
        LOT                       35                  19                  28
        EASEMENT                 NaN                 NaN                 NaN
        OWNER       DESMOND CAMPBELL      CINISOMO MARIO  GANGICHIODO DONALD
        BLDGCL                    B1                  A5                  V0
        TAXCLASS                   1                   1                  1B
        LTFRONT                   18                  25                  16
        LTDEPTH                  100                 100                  19
        STORIES                    2                   3                 NaN
        FULLVAL               407000              415000              128000
        AVLAND                 12337               13301                  81
        AVTOT                  19537               21312                  81
        EXLAND                  1620                1620                   0
        EXTOT                   1620                1620                   0
        EXCD1                   1017                1017                 NaN
        STADDR    140 EAST 49 STREET  537 AMHERST AVENUE        COYLE STREET
        ZIP                    11203               10306                 NaN
        EXMPTCL                   X7                 NaN                 NaN
        BLDFRONT                  18                  14                   0
        BLDDEPTH                  36                  51                   0
        AVLAND2                  NaN                 NaN                 NaN
        AVTOT2                   NaN                 NaN                 NaN
        EXLAND2                  NaN                 NaN                 NaN
        EXTOT2                   NaN                 NaN                 NaN
```

```
EXCD2                    NaN                 NaN                 NaN
PERIOD                 FINAL               FINAL               FINAL
YEAR                 2010/11             2010/11             2010/11
VALTYPE                AC-TR               AC-TR               AC-TR

                      3               4                 5  \
RECORD                4               5                 6
BBLE         4027980132     1006950027E        4031810007
BLOCK              2798             695              3181
LOT                 132              27                 7
EASEMENT            NaN               E               NaN
OWNER              DCAS         CONRAIL  BERGERSON ERIC W
BLDGCL               V0              U6                A5
TAXCLASS             1B               3                 1
LTFRONT              21               0                20
LTDEPTH              75               0               100
STORIES             NaN             NaN                 2
FULLVAL          112613               0            582000
AVLAND             1940               0             17802
AVTOT              1940               0             29859
EXLAND                0               0                 0
EXTOT                 0               0                 0
EXCD1               NaN             NaN               NaN
STADDR    MAZEAU STREET  WEST 23 STREET   90-07 68 AVENUE
ZIP                 NaN             NaN             11375
EXMPTCL             NaN             NaN               NaN
BLDFRONT              0               0                20
BLDDEPTH              0               0                37
AVLAND2             NaN             NaN               NaN
AVTOT2              NaN             NaN               NaN
EXLAND2             NaN             NaN               NaN
EXTOT2              NaN             NaN               NaN
EXCD2               NaN             NaN               NaN
PERIOD            FINAL           FINAL             FINAL
YEAR            2010/11         2010/11           2010/11
VALTYPE           AC-TR           AC-TR             AC-TR

                                6                    7                  8
RECORD                          7                    8                  9
BBLE                   4051861001           3082020064         4052570008
BLOCK                        5186                 8202               5257
LOT                          1001                   64                  8
EASEMENT                      NaN                  NaN                NaN
OWNER            GOLDEN HUANG LLC      SPICER, CLINTON  SILVIA SIPAVICIUS
BLDGCL                         R5                   B1                 A1
TAXCLASS                        4                    1                  1
LTFRONT                         0                   24                 40
LTDEPTH                         0                  100                 96
```

```
STORIES                         6                     2                  2
FULLVAL                    539000                416000             660000
AVLAND                      30960                 13966              14418
AVTOT                      242550                 22345              38064
EXLAND                          0                     0                  0
EXTOT                           0                     0                  0
EXCD1                         NaN                   NaN                NaN
STADDR    43-55 KISSENA BOULEVARD  1200 EAST 95 STREET  172-16 33 AVENUE
ZIP                         11355                 11236              11358
EXMPTCL                       NaN                   NaN                NaN
BLDFRONT                        0                    20                 21
BLDDEPTH                        0                    44                 49
AVLAND2                     30960                   NaN                NaN
AVTOT2                     268740                   NaN                NaN
EXLAND2                       NaN                   NaN                NaN
EXTOT2                        NaN                   NaN                NaN
EXCD2                         NaN                   NaN                NaN
PERIOD                      FINAL                 FINAL              FINAL
YEAR                      2010/11               2010/11            2010/11
VALTYPE                     AC-TR                 AC-TR              AC-TR

                            9
RECORD                     10
BBLE               3070780050
BLOCK                    7078
LOT                        50
EASEMENT                  NaN
OWNER         ABHAS CHAUDHURI
BLDGCL                     C0
TAXCLASS                    1
LTFRONT                    24
LTDEPTH                   100
STORIES                     2
FULLVAL                702000
AVLAND                  18091
AVTOT                   29672
EXLAND                   1620
EXTOT                    1620
EXCD1                    1017
STADDR    1983 WEST 11 STREET
ZIP                     11223
EXMPTCL                   NaN
BLDFRONT                   18
BLDDEPTH                   65
AVLAND2                   NaN
AVTOT2                    NaN
EXLAND2                   NaN
EXTOT2                    NaN
```

```
        EXCD2                     NaN
        PERIOD                  FINAL
        YEAR                  2010/11
        VALTYPE                 AC-TR

In [6]: mydata.describe()

/Users/stevecoggeshall/anaconda3/lib/python3.5/site-packages/numpy/lib/function_bas
  RuntimeWarning)


Out[6]:              RECORD         BLOCK           LOT       LTFRONT       LTDEPTH
        count  1.048575e+06  1.048575e+06  1.048575e+06  1.048575e+06  1.048575e+06
        mean   5.242880e+05  4.708867e+03  3.700924e+02  3.617425e+01  8.827643e+01
        std    3.026977e+05  3.699547e+03  8.605382e+02  7.373356e+01  7.547885e+01
        min    1.000000e+00  1.000000e+00  1.000000e+00  0.000000e+00  0.000000e+00
        25%    2.621445e+05  1.534000e+03  2.300000e+01  1.900000e+01  8.000000e+01
        50%    5.242880e+05  3.944000e+03  4.900000e+01  2.500000e+01  1.000000e+02
        75%    7.864315e+05  6.797000e+03  1.460000e+02  4.000000e+01  1.000000e+02
        max    1.048575e+06  1.635000e+04  9.978000e+03  9.999000e+03  9.999000e+03

                     STORIES       FULLVAL        AVLAND         AVTOT         EXLAN
        count  996433.000000  1.048575e+06  1.048575e+06  1.048575e+06  1.048575e+0
        mean        5.063363  8.804877e+05  8.599503e+04  2.307582e+05  3.681179e+0
        std         8.431372  1.170293e+07  4.100755e+06  6.951206e+06  4.024330e+0
        min         1.000000  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+0
        25%              NaN  3.030000e+05  9.160000e+03  1.838500e+04  0.000000e+0
        50%              NaN  4.460000e+05  1.364600e+04  2.533900e+04  1.620000e+0
        75%              NaN  6.190000e+05  1.970600e+04  4.609500e+04  1.620000e+0
        max       119.000000  6.150000e+09  2.668500e+09  4.668309e+09  2.668500e+0

                       EXTOT          EXCD1           ZIP      BLDFRONT       BLDDEPT
        count  1.048575e+06  622642.000000  1.022219e+06  1.048575e+06  1.048575e+0
        mean   9.254381e+04    1604.500100  1.093532e+04  2.301872e+01  4.007421e+0
        std    6.578281e+06    1388.131676  5.265759e+02  3.578847e+01  4.303640e+0
        min    0.000000e+00    1010.000000  1.000100e+04  0.000000e+00  0.000000e+0
        25%    0.000000e+00            NaN           NaN  1.500000e+01  2.600000e+0
        50%    1.620000e+03            NaN           NaN  2.000000e+01  3.900000e+0
        75%    2.090000e+03            NaN           NaN  2.400000e+01  5.100000e+0
        max    4.668309e+09    7170.000000  3.380300e+04  7.575000e+03  9.393000e+0

                     AVLAND2        AVTOT2       EXLAND2        EXTOT2         EXCD2
        count  2.809660e+05  2.809720e+05  8.667500e+04  1.299330e+05  90941.000000
        mean   2.463655e+05  7.160787e+05  3.518022e+05  6.581148e+05   1371.659098
        std    6.199390e+06  1.169017e+07  1.085248e+07  1.612981e+07   1105.489791
        min    3.000000e+00  3.000000e+00  1.000000e+00  7.000000e+00   1011.000000
        25%             NaN           NaN           NaN           NaN           NaN
        50%             NaN           NaN           NaN           NaN           NaN
```

```
       75%              NaN           NaN           NaN           NaN           Na
       max     2.371005e+09  4.501180e+09  2.371005e+09  4.501180e+09   7160.00000
```

```
In [7]: mydata.count()
```

```
Out[7]: RECORD      1048575
        BBLE        1048575
        BLOCK       1048575
        LOT         1048575
        EASEMENT       4043
        OWNER       1017492
        BLDGCL      1048575
        TAXCLASS    1048575
        LTFRONT     1048575
        LTDEPTH     1048575
        STORIES      996433
        FULLVAL     1048575
        AVLAND      1048575
        AVTOT       1048575
        EXLAND      1048575
        EXTOT       1048575
        EXCD1        622642
        STADDR      1047934
        ZIP         1022219
        EXMPTCL       14992
        BLDFRONT    1048575
        BLDDEPTH    1048575
        AVLAND2      280966
        AVTOT2       280972
        EXLAND2       86675
        EXTOT2       129933
        EXCD2         90941
        PERIOD      1048575
        YEAR        1048575
        VALTYPE     1048575
        dtype: int64
```

```
In [8]: mydata['RECORD'].unique()
```

```
Out[8]: array([      1,       2,       3, ..., 1048573, 1048574, 1048575])
```

```
In [9]: len(mydata['RECORD'])
```

```
Out[9]: 1048575
```

```
In [10]: mydata.set_index('RECORD', inplace = True)
```

```
In [11]: len(mydata['BBLE'].unique())
```

```
Out[11]: 1048575
```

```
In [12]: mydata['BBLE'].head()

Out[12]: RECORD
         1     3046020035
         2     5046820019
         3     3074790028
         4     4027980132
         5    1006950027E
         Name: BBLE, dtype: object

In [13]: mydata['BLOCK'].count() * 100 / numrecords

Out[13]: 100.0

In [14]: len(mydata['BLOCK'].unique())

Out[14]: 13949

In [15]: mydata['BLOCK'].value_counts()

Out[15]: 3944     3888
         16       3786
         3943     3424
         3938     2794
         1171     2535
         3937     2275
         1833     1774
         2450     1651
         1047     1480
         7279     1302
         5893     1295
         8720     1281
         936      1151
         1115     1090
         1320     1049
         1140     1017
         1011      991
         943       946
         1116      881
         1515      869
         3432      853
         1537      842
         1040      821
         870       809
         1536      796
         1165      762
         1048      753
         5137      744
         1373      736
         1419      712
                  ...
         13381       1
         15883       1
         15941       1
         10037       1
         15942       1
         13261       1
         15820       1
         11982       1
         10593       1
         10944       1
         15884       1
         10093       1
         15948       1
         15303       1
         12229       1
         13331       1
         15947       1
         9067        1
         15885       1
         10825       1
         14009       1
         15936       1
         16324       1
         11340       1
         12230       1
         9664        1
         10688       1
         7529        1
         9665        1
         6594        1
         Name: BLOCK, dtype: int64
```

```
In [16]: mydata['BLOCK'].min()
```

```
Out[16]: 1
```

```
In [17]: mydata['LOT'].count() * 100 / numrecords
```

```
Out[17]: 100.0
```

```
In [18]: len(mydata['LOT'].unique())
```

```
Out[18]: 6366
```

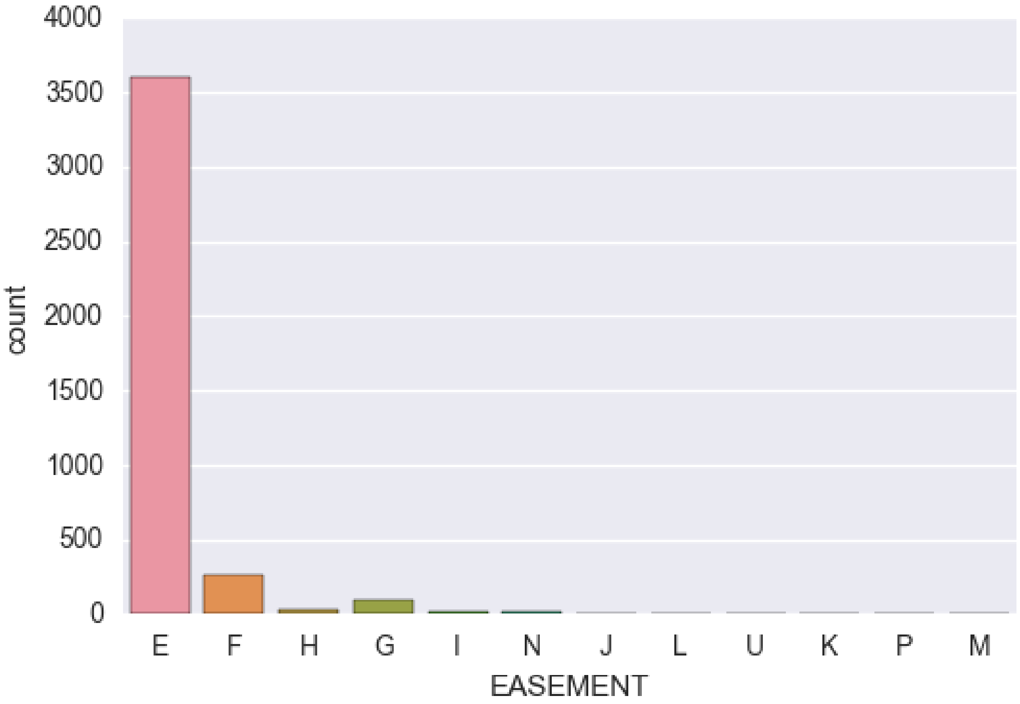
```
In [19]: mydata['LOT'].value_counts()
```

```
Out[19]: 1        23570
         20       12045
```

```
15      11904
12      11894
14      11864
16      11810
18      11763
17      11728
25      11692
21      11593
23      11469
22      11462
6       11418
19      11408
24      11392
26      11390
30      11354
28      11170
29      11149
27      11107
13      11086
7       11070
10      10876
9       10872
11      10773
8       10673
32      10616
33      10546
31      10502
35      10490
        ...
4902        1
5548        1
5409        1
7217        1
4889        1
7223        1
5401        1
4894        1
6061        1
5406        1
8108        1
4895        1
5407        1
4892        1
6123        1
5404        1
4893        1
5405        1
7216        1
```

```
         4898        1
         5410        1
         4899        1
         5411        1
         8109        1
         4896        1
         5408        1
         6060        1
         4897        1
         7145        1
         6043        1
         Name: LOT, dtype: int64
```

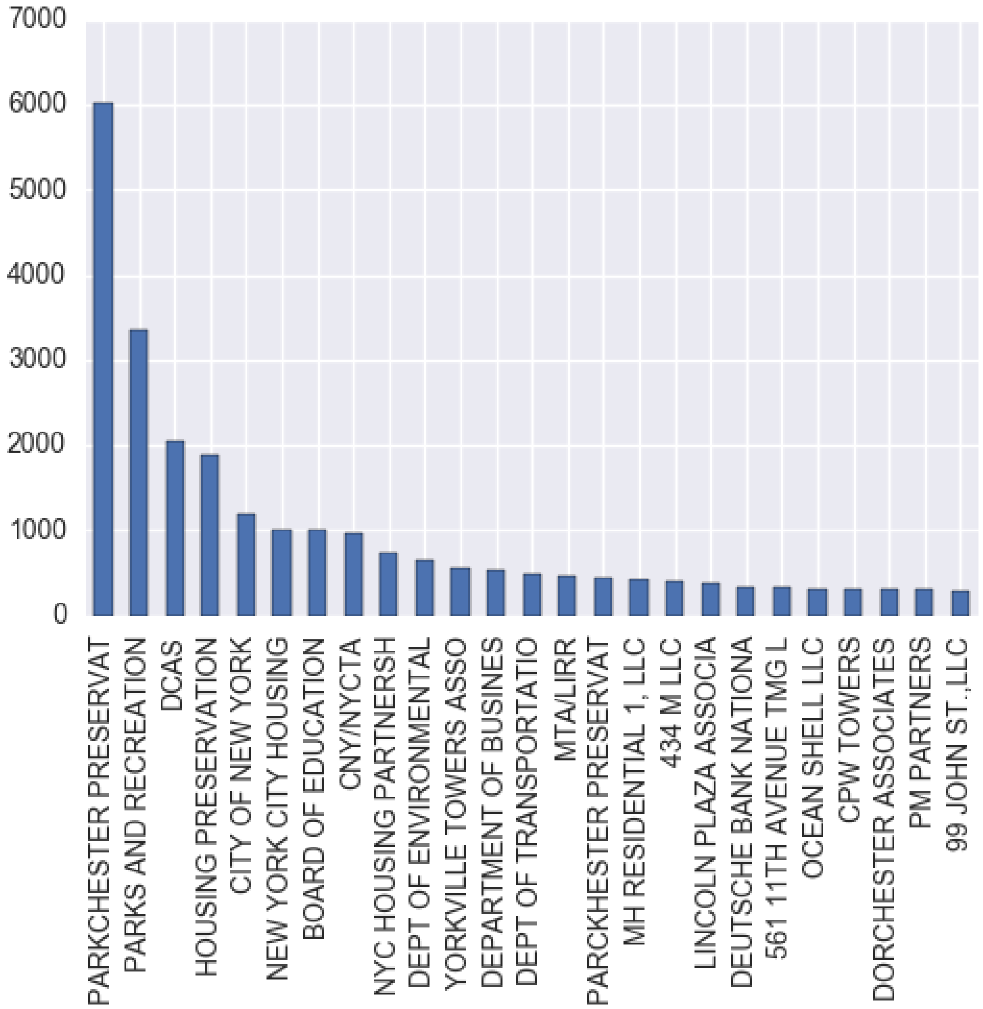
```
In [20]: mydata['LOT'].min()

Out[20]: 1

In [21]: mydata['EASEMENT'].count() * 100 / numrecords

Out[21]: 0.38557089383210547

In [22]: len(mydata['EASEMENT'].unique())

Out[22]: 13

In [23]: mydata['EASEMENT'].value_counts()

Out[23]: E    3603
         F     265
         G      95
         H      30
         N      17
         I      14
         J       7
         K       4
         L       3
         P       2
         M       2
         U       1
         Name: EASEMENT, dtype: int64

In [24]: sns.countplot(x='EASEMENT', data = mydata)
         plt.savefig('hist.png')
```



```
In [25]: mydata['OWNER'].count() * 100 / numrecords

Out[25]: 97.035691295329372

In [26]: len(mydata['OWNER'].unique())

Out[26]: 847054

In [27]: mydata['OWNER'].value_counts()

Out[27]: PARKCHESTER PRESERVAT     6021
         PARKS AND RECREATION      3358
         DCAS                      2053
         HOUSING PRESERVATION      1900
         CITY OF NEW YORK          1189
         NEW YORK CITY HOUSING     1014
         BOARD OF EDUCATION        1003
         CNY/NYCTA                  975
         NYC HOUSING PARTNERSH      747
         DEPT OF ENVIRONMENTAL      644
         YORKVILLE TOWERS ASSO      558
         DEPARTMENT OF BUSINES      526
         DEPT OF TRANSPORTATIO      484
         MTA/LIRR                   467
         PARCKHESTER PRESERVAT      439
```

```
MH RESIDENTIAL 1, LLC     411
434 M LLC                 393
LINCOLN PLAZA ASSOCIA     366
DEUTSCHE BANK NATIONA     333
561 11TH AVENUE TMG L     324
OCEAN SHELL LLC           314
CPW TOWERS                314
DORCHESTER ASSOCIATES     313
PM PARTNERS               301
99 JOHN ST.,LLC           296
NEW YORK CITY TRANSIT     271
FIRE DEPARTMENT           249
TRUSTEES OF COLUMBIA      239
BRIGHTWATER TOWERS        222
POLICE DEPARTMENT         212
                         ...
ARNALDO PEREZ               1
MERCADO NELSON              1
ANTHONY FOSTER              1
BARBER, JOSEPH              1
HERMAN PARDO                1
A CORVI                     1
SATO, MAIKO                 1
HOOSEIN, AYUBE              1
SORGINI, ASSUNTA            1
MARTIN MAUREEN H            1
156 WEST 74TH STREET,       1
OK FURNITURE LIQUIDAT       1
845-855 DEAN STREET         1
ALEXANDER, ALTHEA L         1
NASPUD-ACERO, SEGUNDO       1
ARENA JOHN                  1
SOLOMON MELZER              1
JOYCE THOMAS E & NUAL       1
NICHOLAS PURPERO            1
95 SO 5TH ST CORP           1
MITA TADEUSZ HJ             1
MORRISON, ANITA             1
RANOLA, JUNE M              1
11 WEST END AVENUE, L       1
CHUI, KIN KEUNG             1
FRANK FARGIANO              1
YAO, JORDAN ZHI HUA         1
PERSAUD, PRAIMANANDA        1
SPEARS, JOSHUA J            1
SANCHEZ, VIRGILIO           1
Name: OWNER, dtype: int64
```

```
In [28]: mydata['OWNER'].value_counts().head(25).plot(kind='bar')

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x122d15748>
```



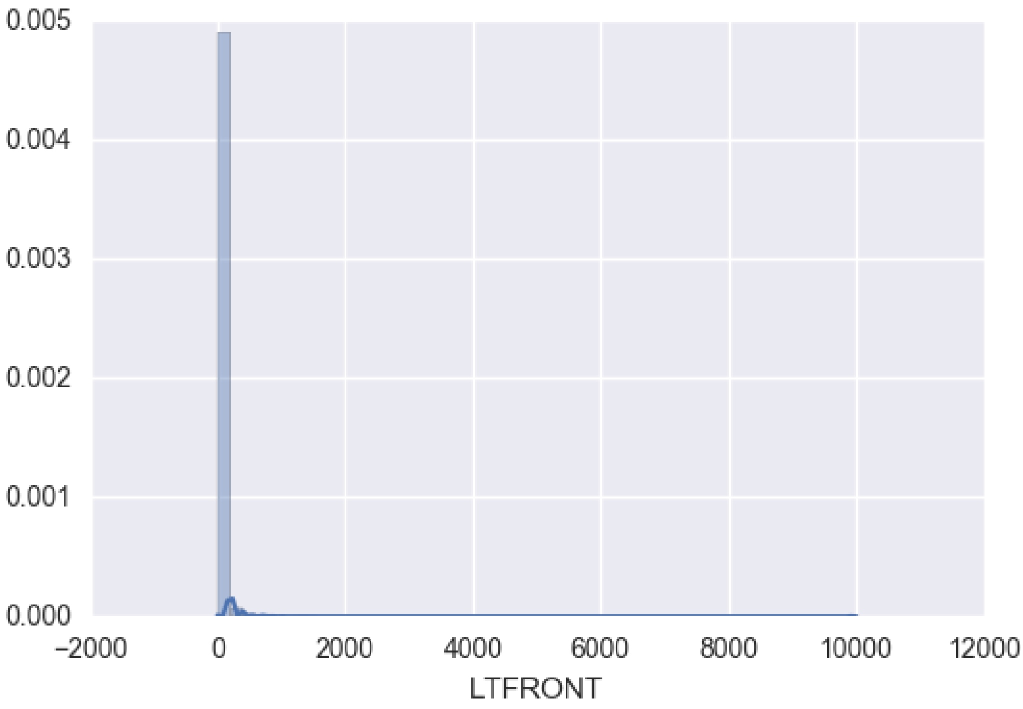
```
In [29]: mydata['BLDGCL'].count() * 100 / numrecords

Out[29]: 100.0

In [30]: len(mydata['BLDGCL'].unique())

Out[30]: 200

In [31]: mydata['BLDGCL'].value_counts()

Out[31]: R4     139879
         A1     119340
```

```
A5     92896
B1     84054
B2     73156
C0     73077
B3     59091
A2     49085
A9     25931
B9     25235
R5     23950
V0     21520
R3     20899
C3     16332
C1     15070
S2     14480
C2     13632
R2     10558
R1      8015
C7      7544
K1      7529
V1      6836
A0      6815
S1      6133
A3      5742
R0      5681
G7      5562
D1      5101
S9      4120
K9      4051
       ...
H1        28
Y8        27
Q7        27
U0        25
Y7        25
F8        24
J5        20
J8        18
C8        18
V4        17
U5        17
N1        16
Q5        16
T1        15
Y3        15
P4        14
V6        14
P1        13
J1        12
```

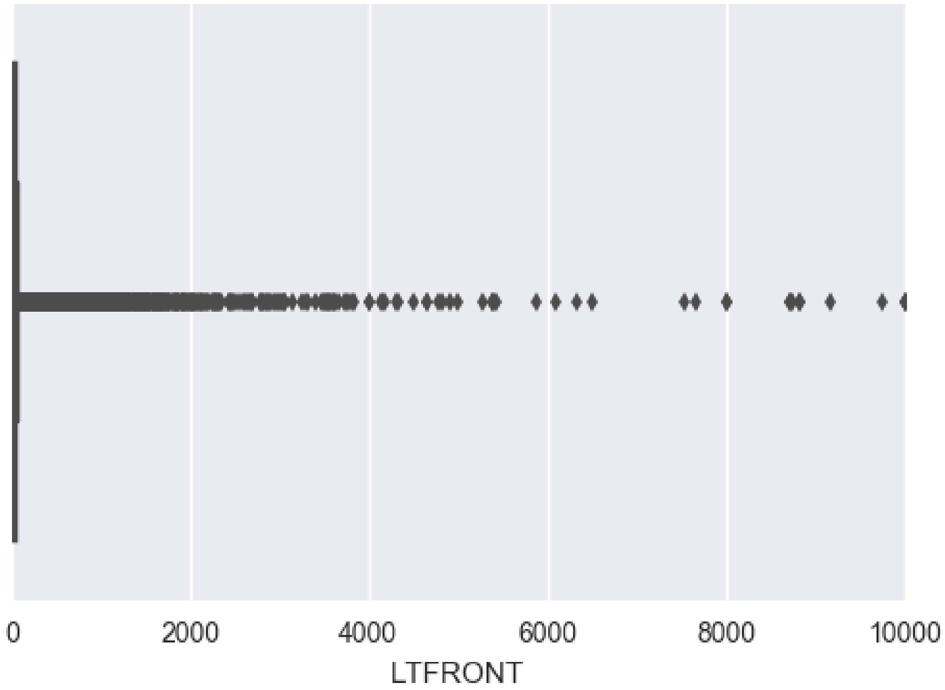
```
         N3        11
         J3         8
         J7         8
         N4         7
         J2         7
         Z5         6
         I3         4
         I2         4
         H7         3
         E6         1
         Y5         1
         Name: BLDGCL, dtype: int64

In [32]: mydata[mydata['BLDGCL'] == 0]

Out[32]: Empty DataFrame
         Columns: [BBLE, BLOCK, LOT, EASEMENT, OWNER, BLDGCL, TAXCLASS, LTFRONT, LT
         Index: []

         [0 rows x 29 columns]

In [33]: mydata['TAXCLASS'].count() * 100 / numrecords

Out[33]: 100.0

In [34]: len(mydata['TAXCLASS'].unique())

Out[34]: 11

In [35]: mydata['TAXCLASS'].value_counts()

Out[35]: 1     643774
         2     188592
         4     102281
         2A     40558
         1B     22193
         1A     20899
         2B     13962
         2C     10795
         3       4546
         1C       946
         1D        29
         Name: TAXCLASS, dtype: int64

In [36]: mydata['LTFRONT'].count() * 100 / numrecords

Out[36]: 100.0

In [37]: sns.distplot(mydata['LTFRONT'])
```

```
/Users/stevecoggeshall/anaconda3/lib/python3.5/site-packages/statsmodels/nonparamet
  y = X[:m/2+1] + np.r_[0,X[m/2+1:],0]*1j
```

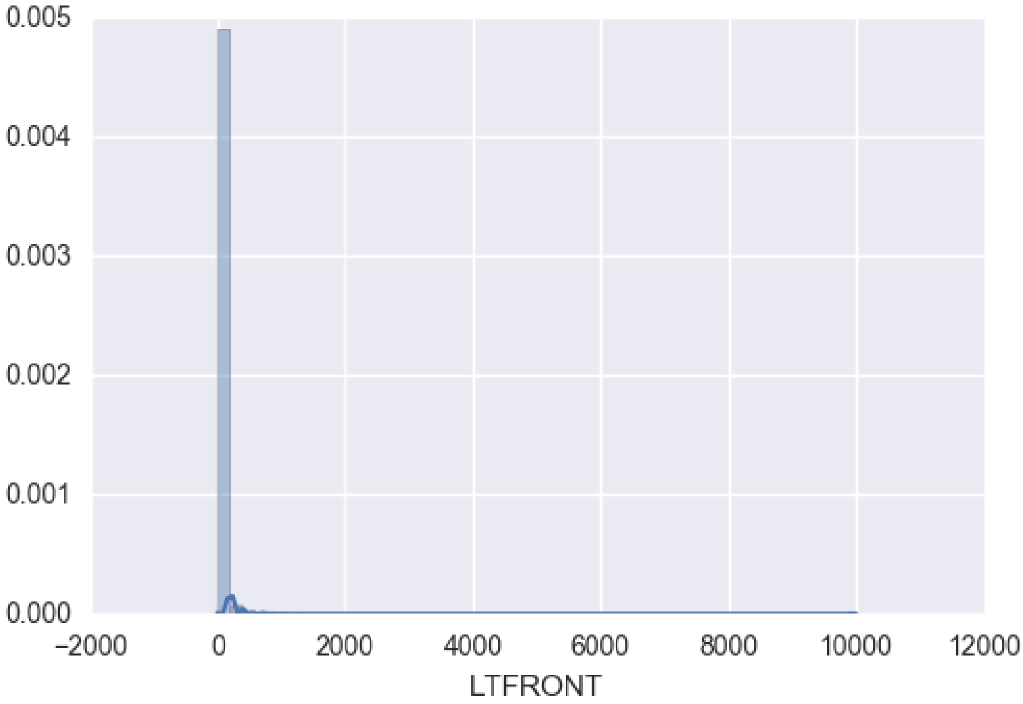
```
Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x11eb8bda0>
```



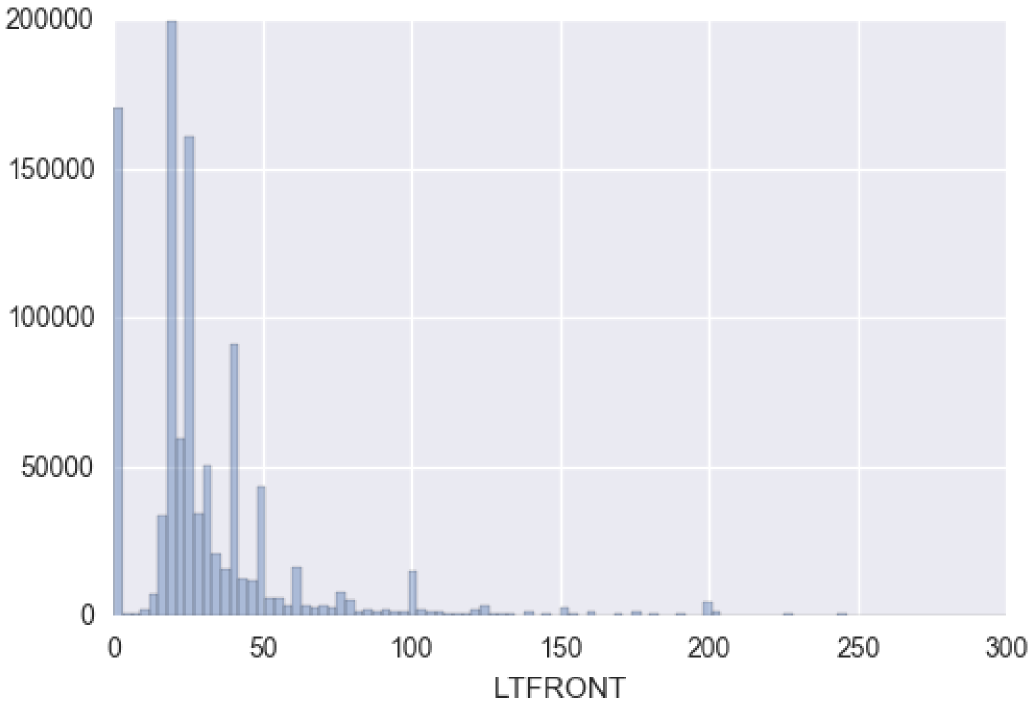
```
In [38]: sns.boxplot(x='LTFRONT', data=mydata)

Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x119785978>
```



```
In [39]: sns.distplot(mydata['LTFRONT'])

/Users/stevecoggeshall/anaconda3/lib/python3.5/site-packages/statsmodels/nonparamet
  y = X[:m/2+1] + np.r_[0,X[m/2+1:],0]*1j


Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x118d07fd0>
```



```
In [40]: xhigh = 300
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['LTFRONT'] <= xhigh]
         sns.distplot(temp['LTFRONT'],bins=100, kde=False)

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x11ede29e8>
```



```
In [41]: len(mydata[mydata['LTFRONT']==0])

Out[41]: 168867

In [42]: len(mydata[mydata['LTFRONT']==1])

Out[42]: 819

In [43]: len(mydata[mydata["LTFRONT"]==2])

Out[43]: 750

In [44]: mydata['LTFRONT'].value_counts()

Out[44]: 0       168867
         20      134447
         25      116301
         40       81802
         18       40188
         50       38577
         30       35973
         19       25185
         24       25180
         22       23304
         26       19415
```

```
21       19319
16       18359
23       16801
60       13851
100      12991
28       12963
27       12485
17       10372
29        9249
33        8007
37        7904
35        7526
32        7426
31        7243
45        6708
75        6593
41        5929
42        5629
34        5036
         ...
2167         1
1802         1
1307         1
2333         1
2845         1
1311         1
609          1
1325         1
4910         1
811          1
2858         1
612          1
1321         1
809          1
1125         1
1832         1
1320         1
1126         1
2345         1
2662         1
1831         1
1319         1
616          1
806          1
1129         1
1317         1
1130         1
1315         1
```

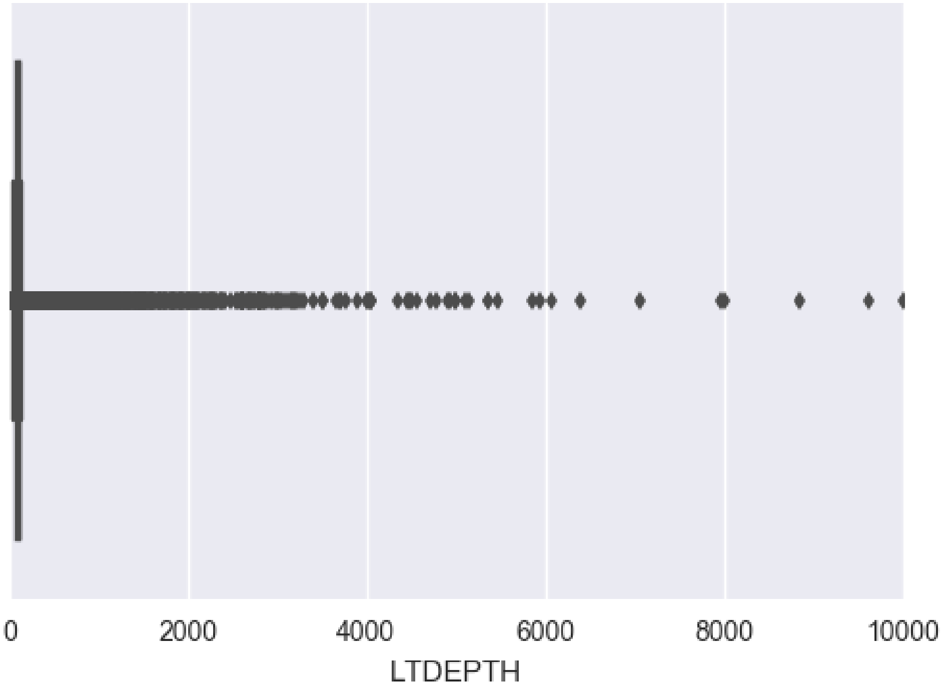
```
        803           1
        1023          1
        Name: LTFRONT, dtype: int64
```

```
In [45]: mydata['LTDEPTH'].count() * 100 / numrecords
```

```
Out[45]: 100.0
```

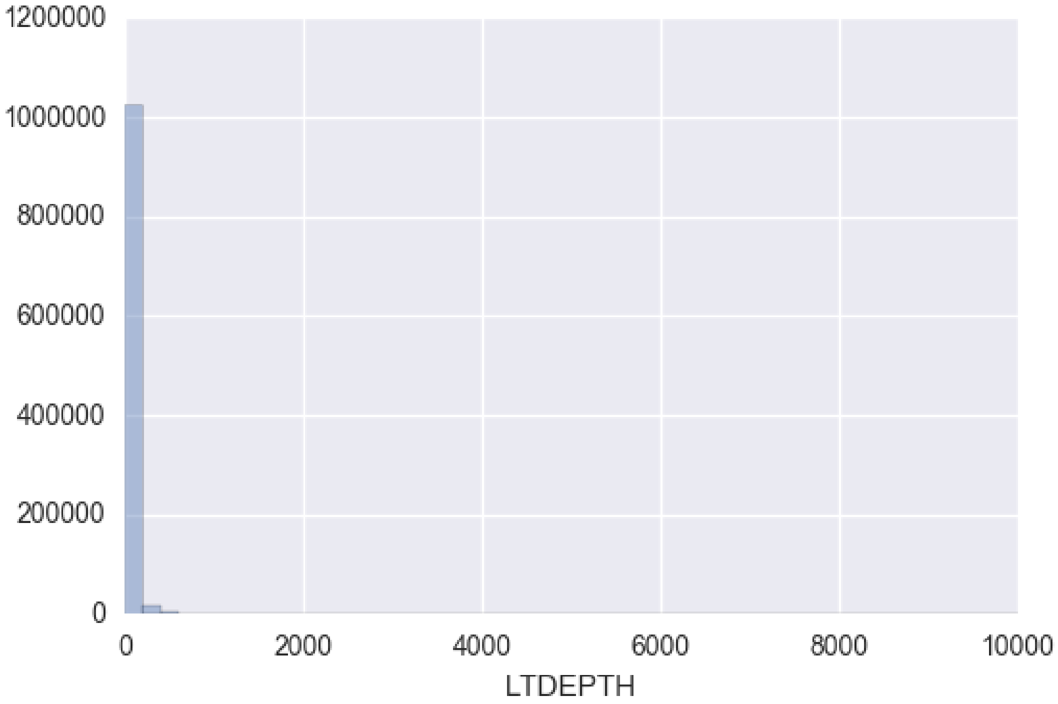
```
In [46]: sns.boxplot(x='LTDEPTH', data=mydata)
```

```
Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x13038fe10>
```

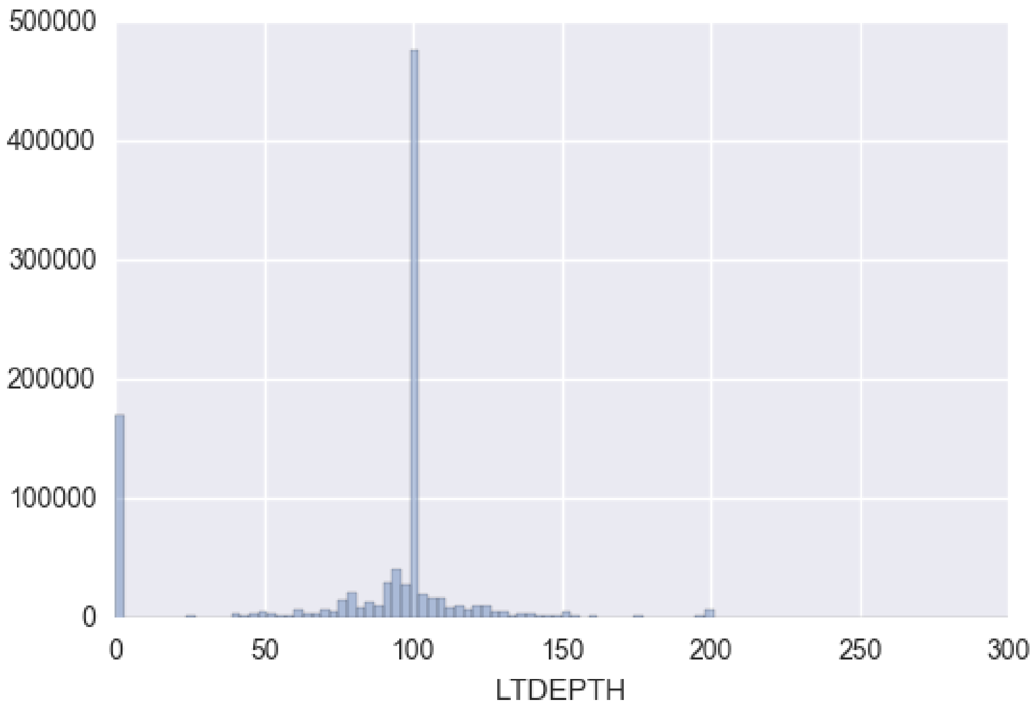


```
In [47]: sns.distplot(mydata['LTDEPTH'],kde=False)
```

```
Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x119637588>
```



```
In [48]: xhigh = 300
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['LTDEPTH'] <= xhigh]
         sns.distplot(temp['LTDEPTH'],bins=100, kde=False)

Out[48]: <matplotlib.axes._subplots.AxesSubplot at 0x11b0d1eb8>
```



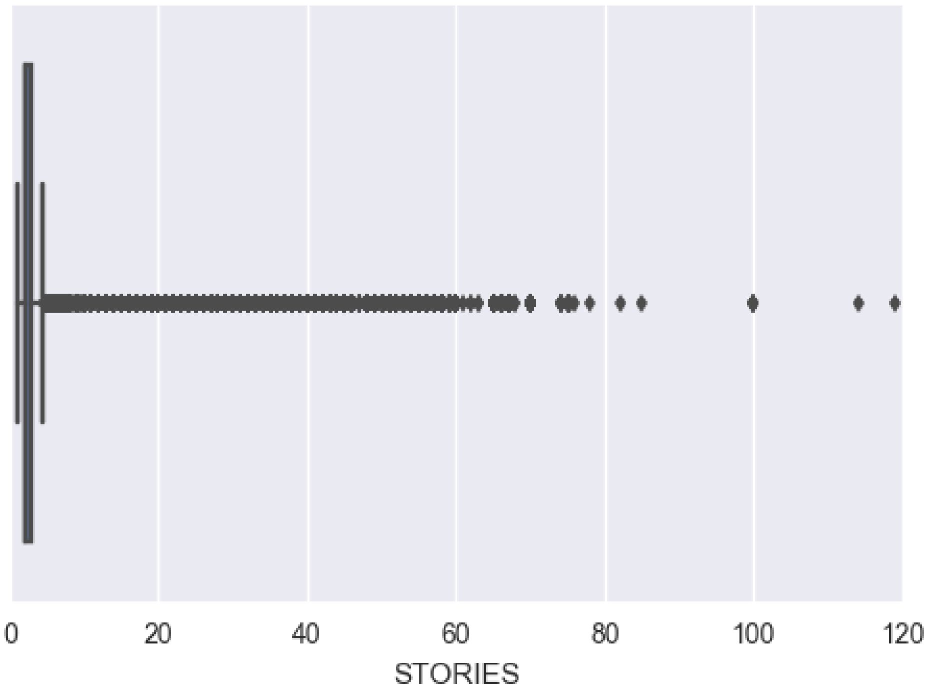
```
In [49]: len(mydata[mydata['LTDEPTH']==0])

Out[49]: 169888

In [50]: len(mydata[mydata['LTDEPTH']==1])

Out[50]: 126

In [51]: len(mydata[mydata["LTDEPTH"]==2])

Out[51]: 79

In [52]: mydata['LTDEPTH'].value_counts()

Out[52]: 100       457583
         0         169888
         95         31022
         90         19941
         80         16414
         99         11133
         75           9969
         97           9839
         102          9377
         96           9154
         110          8555
```

```
98        8515
105       8173
101       7559
120       7328
85        7195
125       7182
103       6780
92        6577
200       6419
94        5797
93        4788
107       4615
60        4419
109       4404
50        4388
87        4341
104       4240
150       4188
114       4116
         ...
1163         1
1399         1
1152         1
1144         1
887          1
1279         1
1909         1
4471         1
882          1
1148         1
1660         1
1905         1
879          1
2175         1
4463         1
1386         1
8847         1
2181         1
1157         1
1158         1
1670         1
869          1
2694         1
647          1
1159         1
1161         1
2400         1
1376         1
```

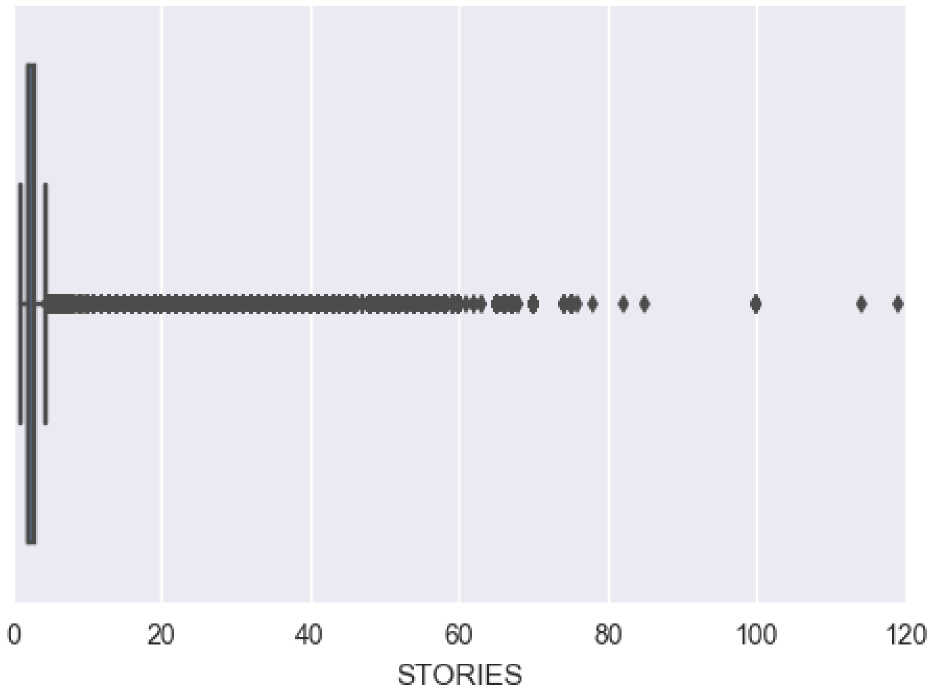
```
          1162          1
          1023          1
          Name: LTDEPTH, dtype: int64
```

```
In [53]: mydata['STORIES'].count() * 100 / numrecords

Out[53]: 95.02734663710273 8
```

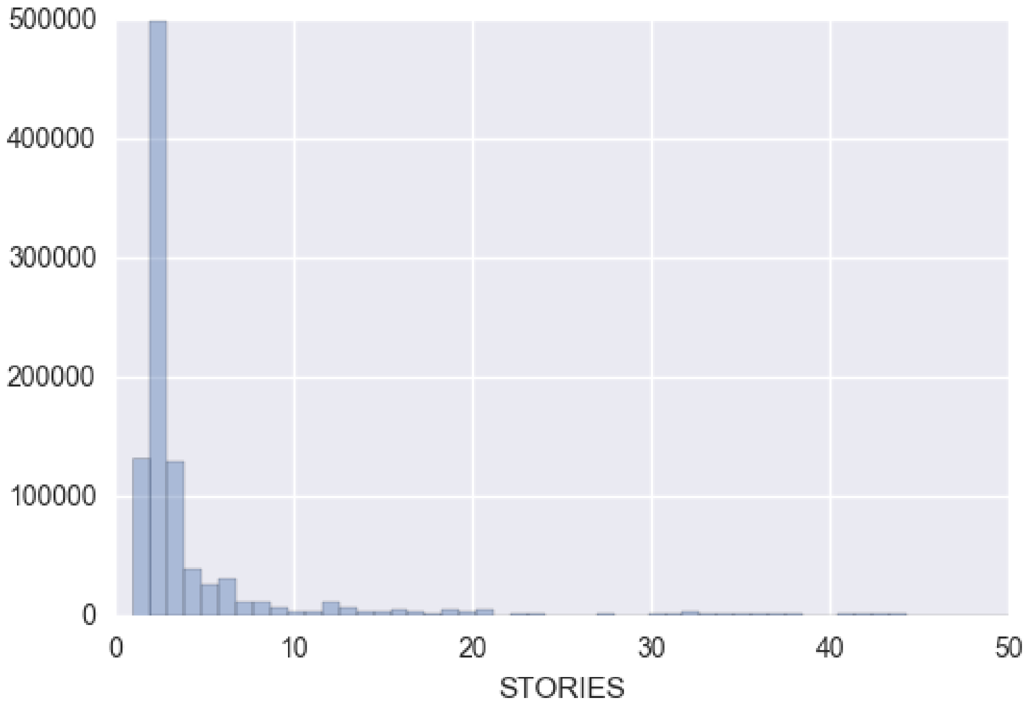
```
In [54]: sum(pd.isnull(mydata['STORIES']))

Out[54]: 52142

In [55]: sns.boxplot(x='STORIES', data=mydata)
         plt.savefig("boxplot.png")
```



```
In [56]: len(mydata[mydata['STORIES'] == 0])

Out[56]: 0

In [57]: sns.boxplot(x='STORIES', data=mydata)

Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x11ed26b00>
```



```
In [58]: xhigh = 50
         temp = mydata[mydata['STORIES'] > 0]
         temp.count()
         sns.plt.xlim(0,xhigh)
         temp = temp[temp['STORIES'] <= xhigh]
         sns.distplot(temp['STORIES'],bins=51, kde=False)

Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x120df50b8>
```



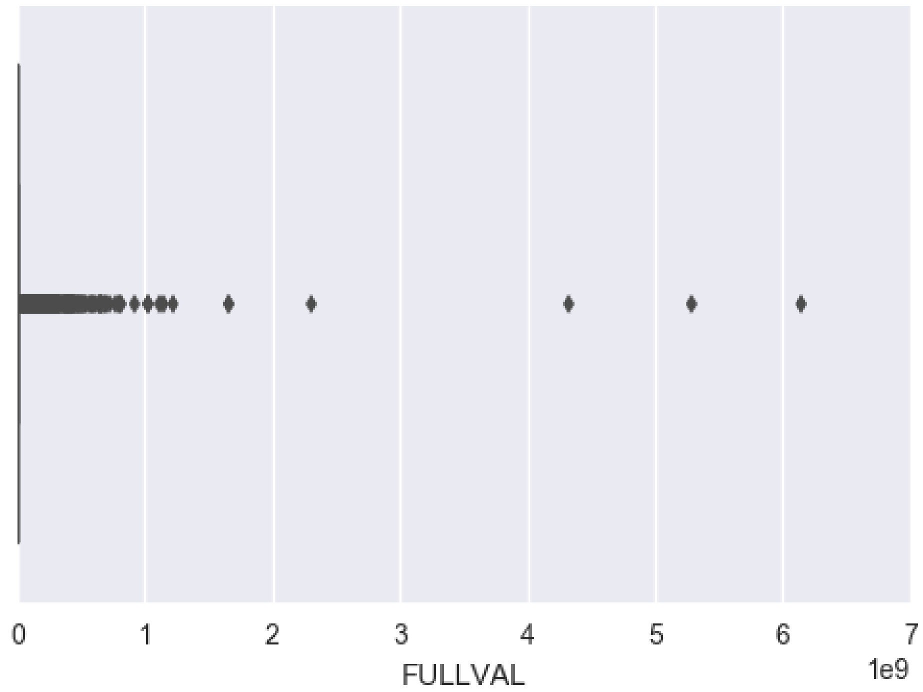
```
In [59]: mydata['STORIES'].value_counts()

Out[59]: 2.0      403318
         3.0      128493
         1.0       93606
         2.5       81304
         4.0       38337
         6.0       30936
         5.0       25971
         1.5       24354
         2.7       13543
         12.0      12198
         8.0       11953
         7.0       11899
         1.6        8816
         9.0        7343
         13.0       7330
         16.0       5428
         1.7        5051
         21.0       4885
         19.0       4866
         11.0       4459
         15.0       4270
         10.0       3758
```

```
17.0       3457
14.0       3368
20.0       3141
32.0       3127
30.0       2905
42.0       2875
31.0       2583
27.0       2333
          ...
6.7          12
59.0         12
3.6          11
47.0         10
4.7          10
1.9          10
74.0          6
100.0         5
3.3           5
9.5           4
1.3           3
1.1           3
8.5           3
62.0          3
2.8           3
5.7           2
1.4           2
2.4           2
68.0          2
63.0          2
2.1           1
4.2           1
82.0          1
114.0         1
85.0          1
78.0          1
76.0          1
61.0          1
2.9           1
119.0         1
Name: STORIES, dtype: int64
```

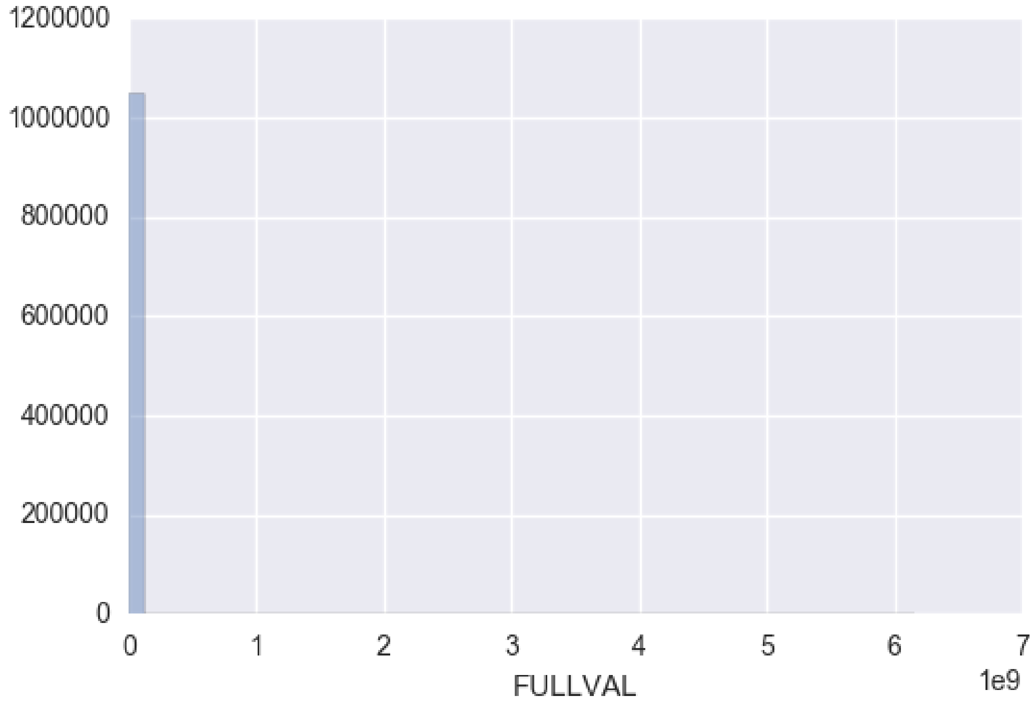
```
In [60]: mydata['FULLVAL'].count() * 100 / numrecords
```

```
Out[60]: 100.0
```

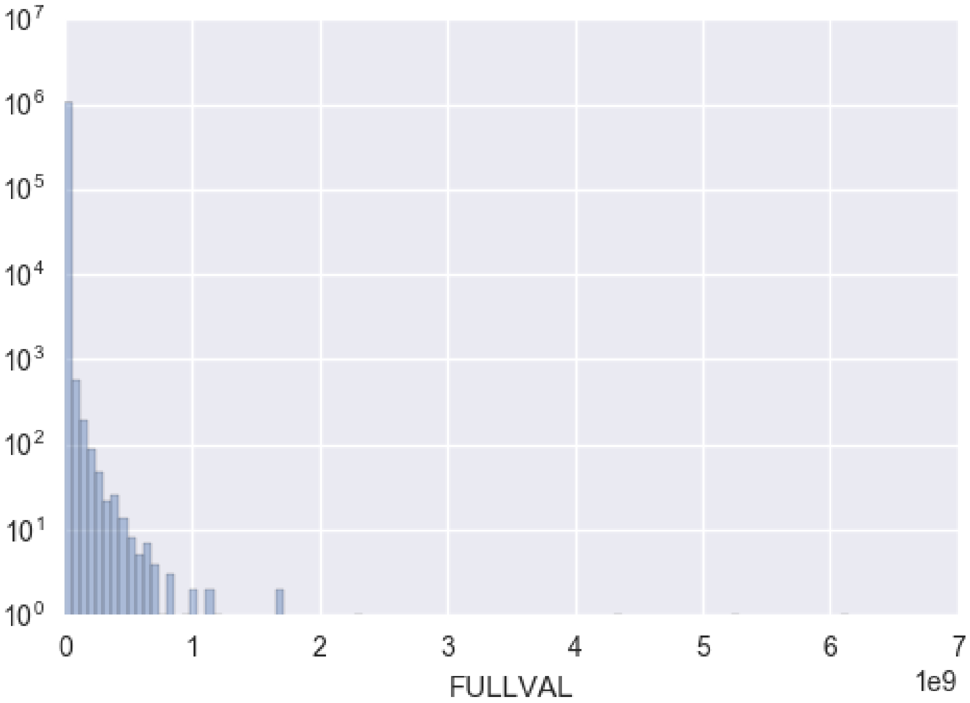
```
In [61]: sns.boxplot(x='FULLVAL', data=mydata)
         plt.savefig("boxplot.png")
```



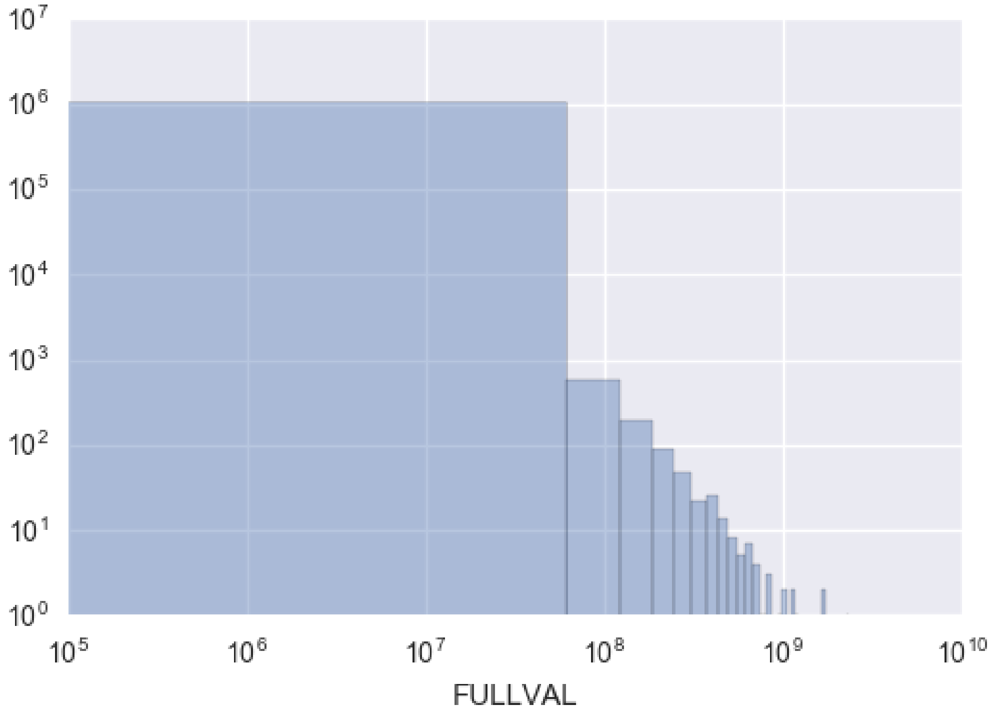
```
In [62]: sns.distplot(mydata['FULLVAL'],kde=False)
         plt.savefig('dist bad.png')
```



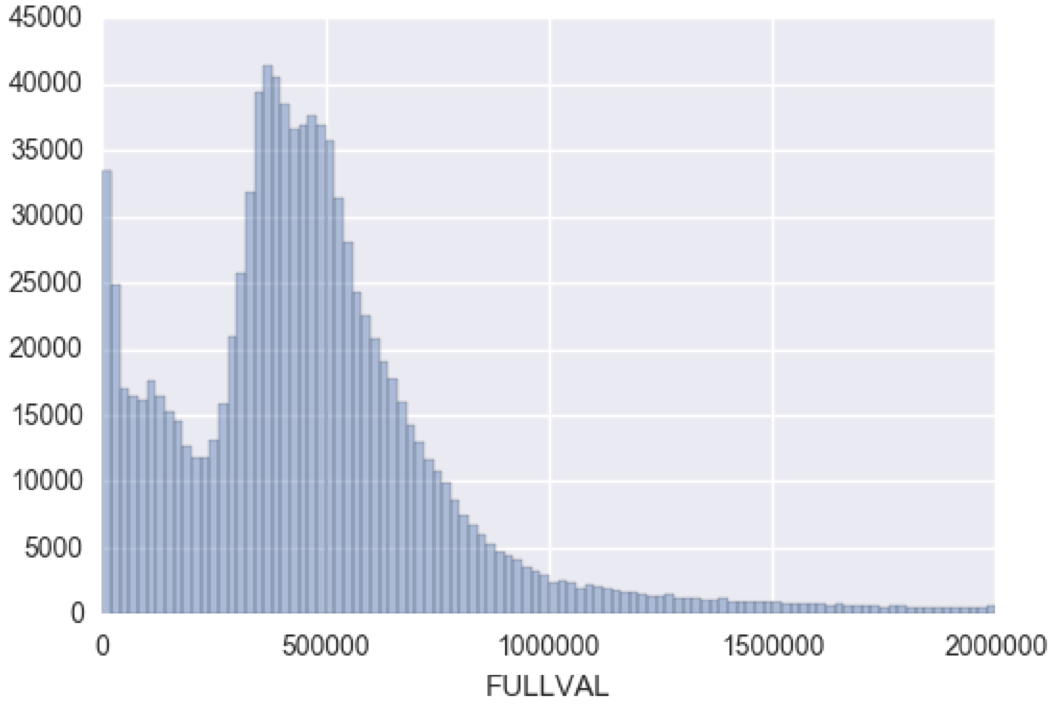
```
In [63]: temp = mydata[mydata['FULLVAL'] >= 0]
         ax = sns.distplot(temp['FULLVAL'],bins=100, kde=False)
         ax.set_yscale('log')
         plt.savefig('log.png')
```



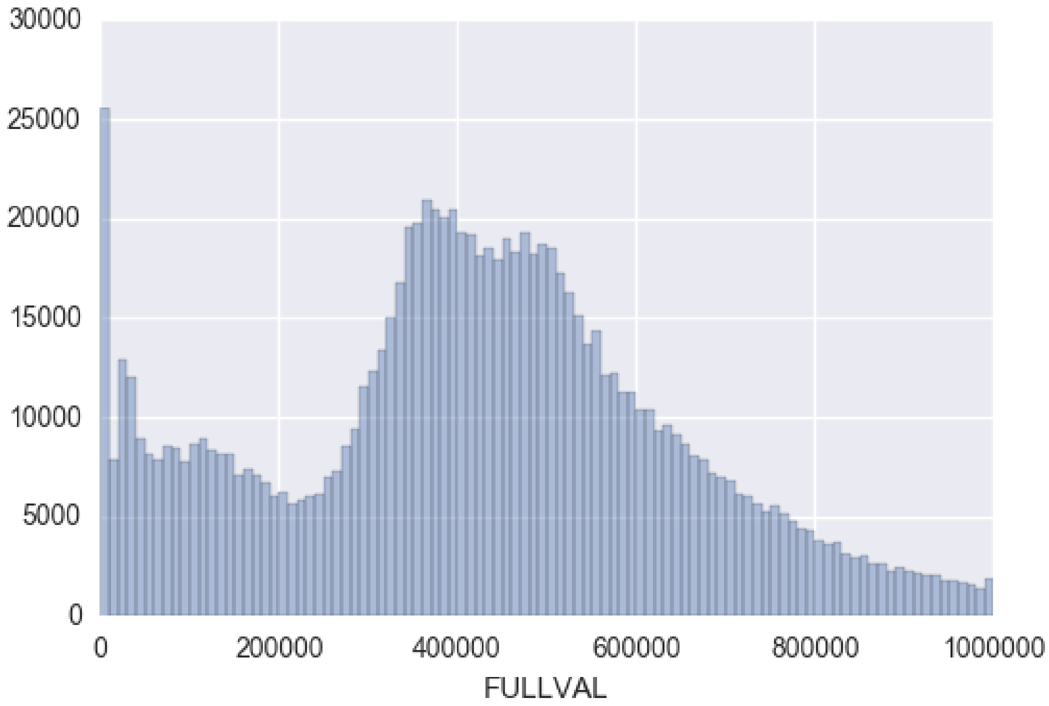
```
In [64]: temp = mydata[mydata['FULLVAL'] >= 0]
         ax = sns.distplot(temp['FULLVAL'],bins=100, kde=False)
         ax.set_yscale('log')
         ax.set_xscale('log')
         plt.savefig('loglog.png')
```



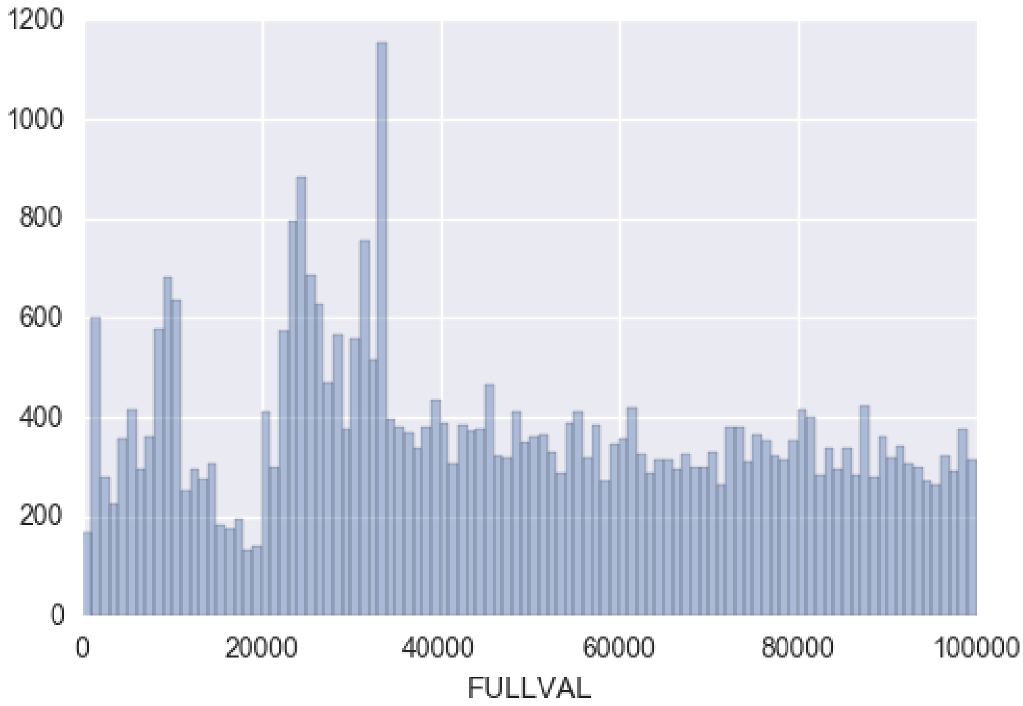
```
In [65]: xhigh = 2000000
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['FULLVAL'] <= xhigh]
         sns.distplot(temp['FULLVAL'],bins=100, kde=False)
         plt.savefig('dist good.png')
```



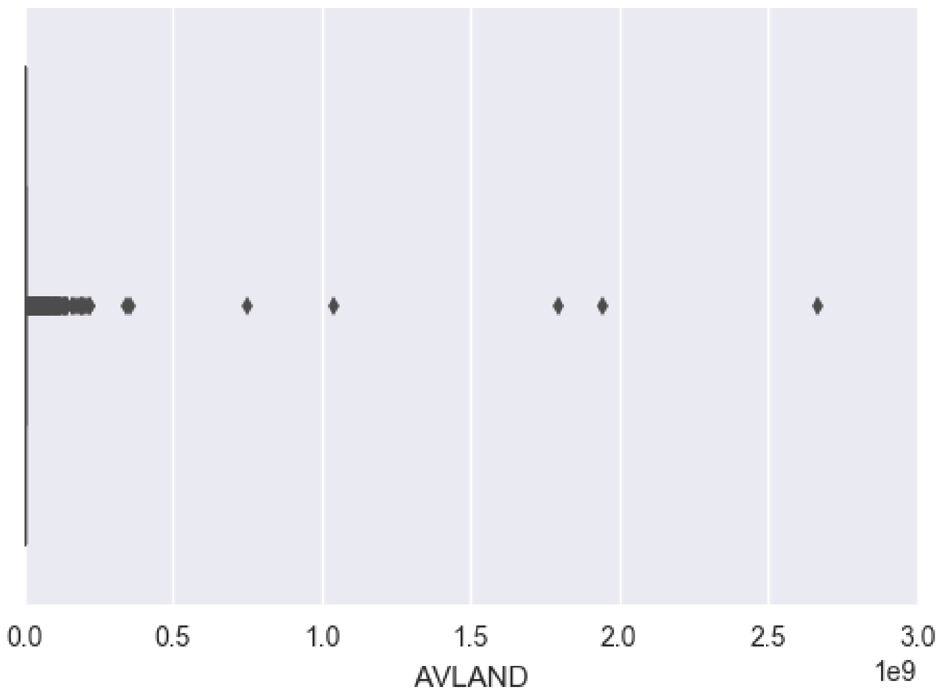
```
In [66]: xhigh = 1000000
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['FULLVAL'] <= xhigh]
         sns.distplot(temp['FULLVAL'],bins=100, kde=False)

Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x1449adcf8>
```



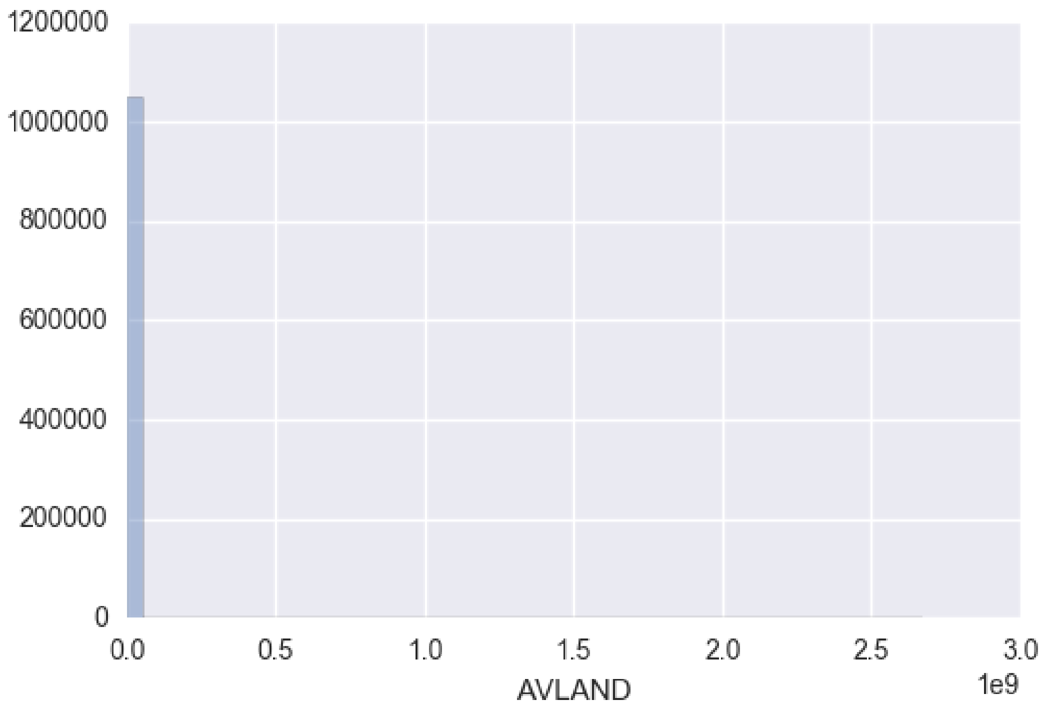
```
In [67]: xhigh = 100000
         sns.plt.xlim(0,xhigh)
         temp = mydata[(mydata['FULLVAL'] <= xhigh) & (mydata['FULLVAL']) > 0]
         sns.distplot(temp['FULLVAL'],bins=100, kde=False)

Out[67]: <matplotlib.axes._subplots.AxesSubplot at 0x118b61c88>
```



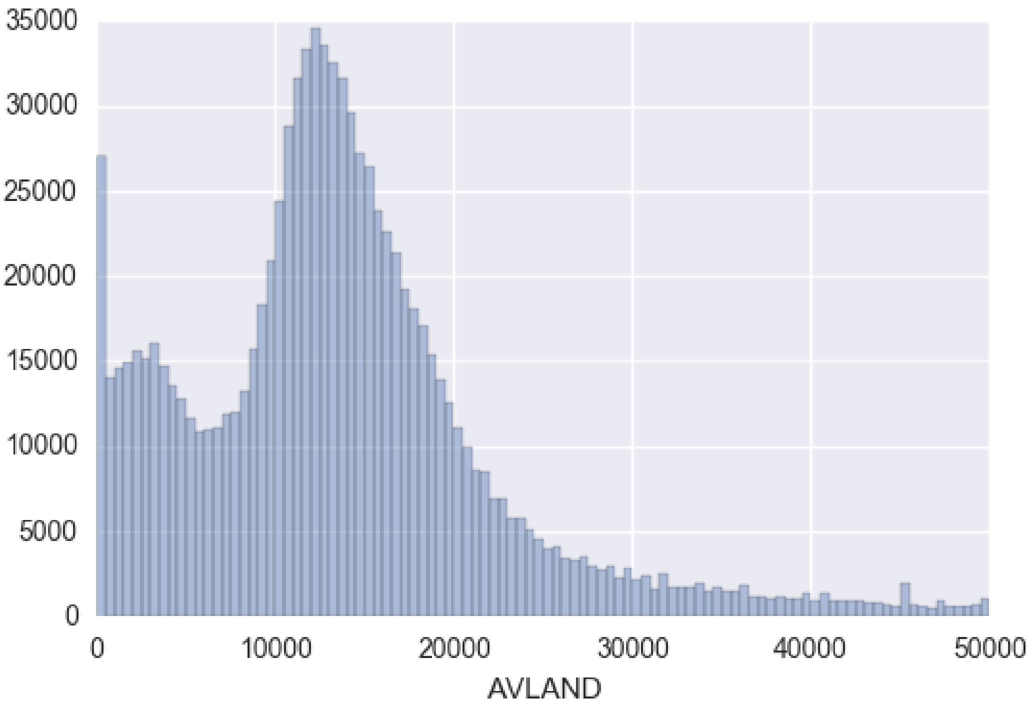
```
In [68]: mydata['AVLAND'].count() * 100 / numrecords

Out[68]: 100.0

In [69]: sns.boxplot(x='AVLAND', data=mydata)

Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x126a96278>
```



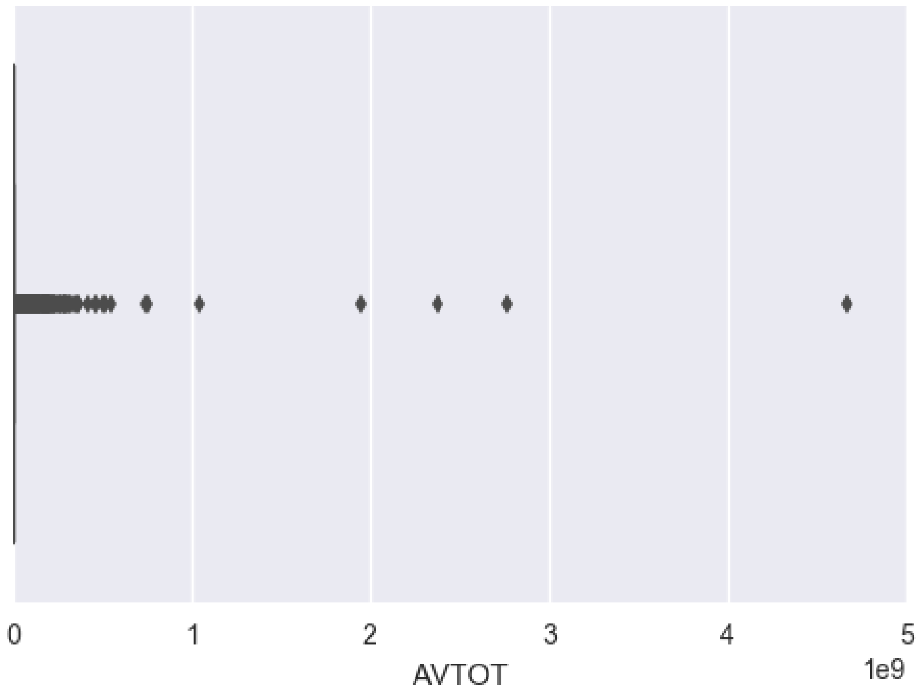
```
In [70]: sns.distplot(mydata['AVLAND'],kde=False)

Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x1296e57b8>
```



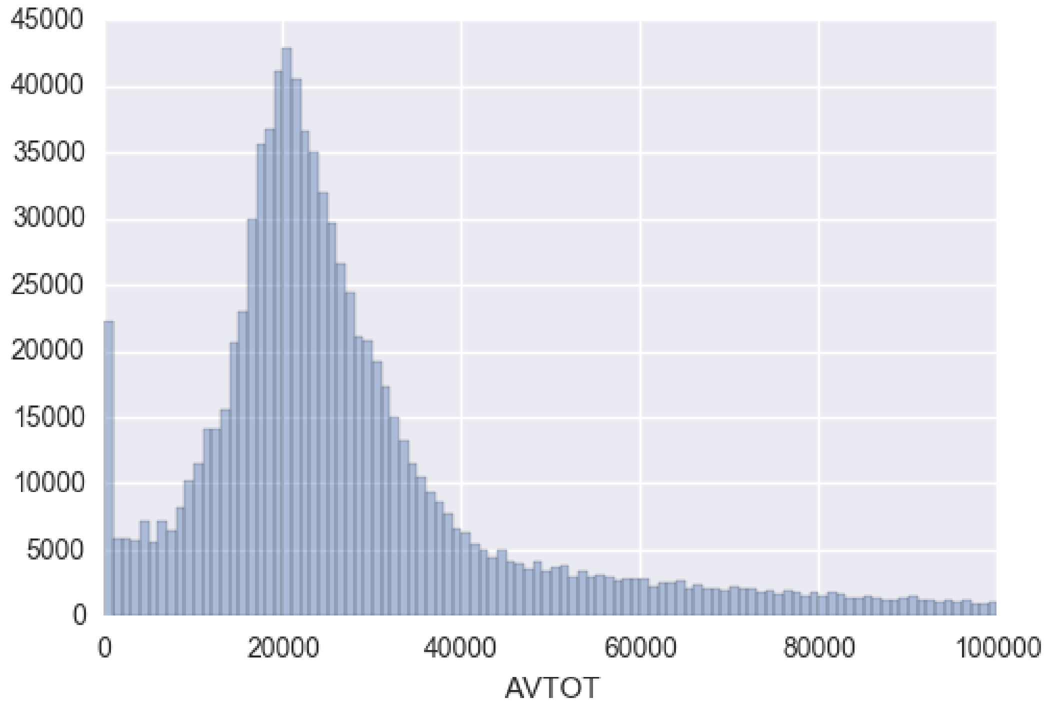
```
In [71]: xhigh = 50000
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['AVLAND'] <= xhigh]
         sns.distplot(temp['AVLAND'],bins=100, kde=False)

Out[71]: <matplotlib.axes._subplots.AxesSubplot at 0x12a4c3400>
```



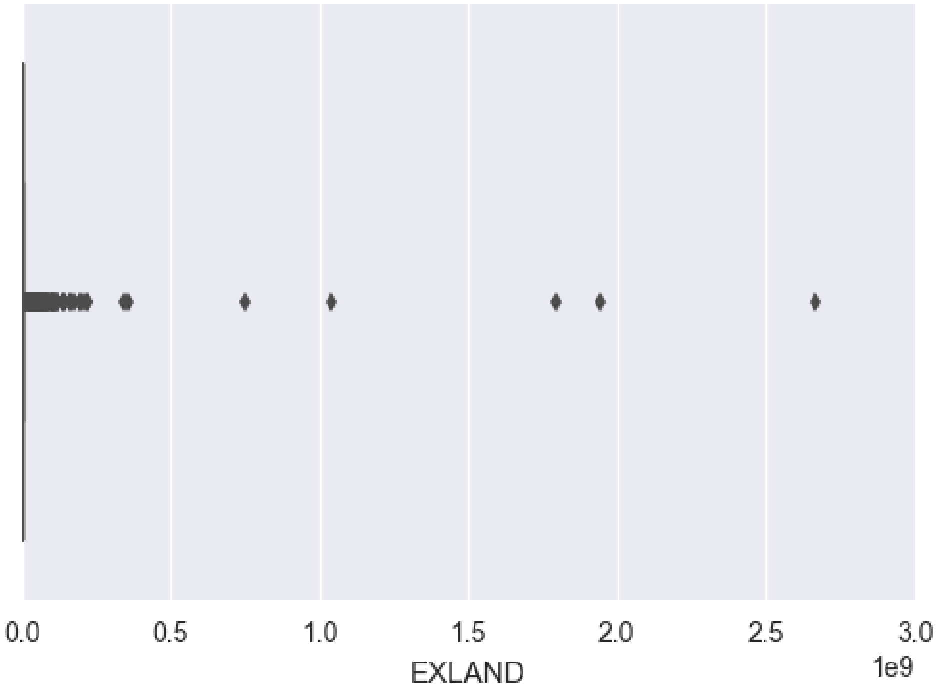
```
In [72]: mydata['AVTOT'].count() * 100 / numrecords

Out[72]: 100.0

In [73]: sns.boxplot(x='AVTOT', data=mydata)

Out[73]: <matplotlib.axes._subplots.AxesSubplot at 0x12c51ea90>
```



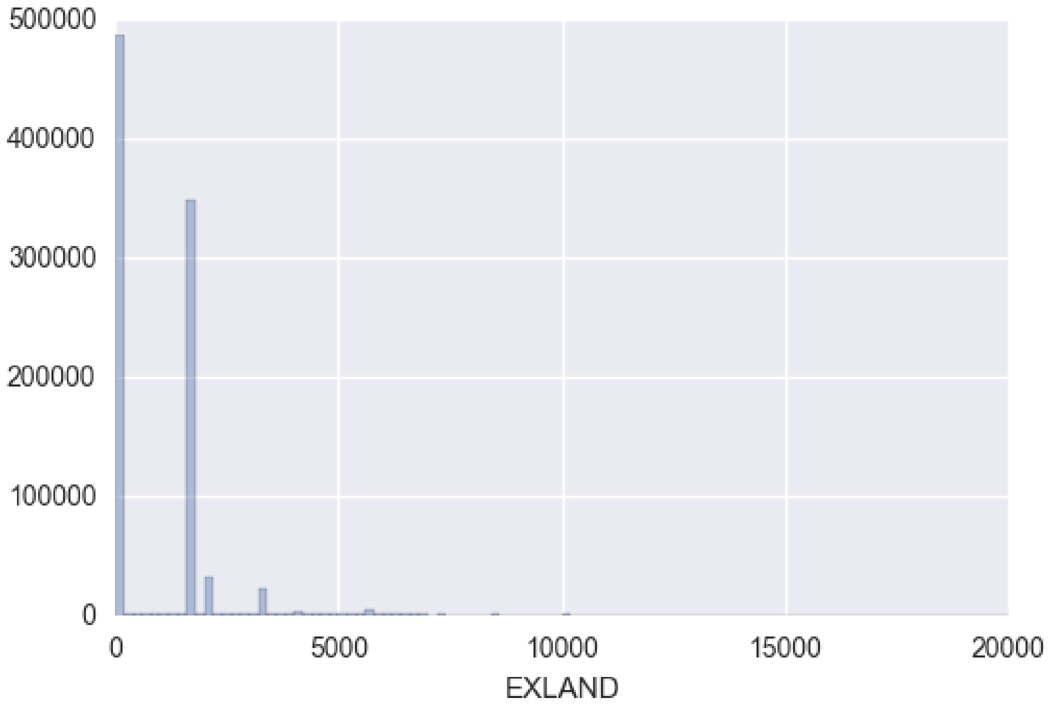
```
In [74]: xhigh = 100000
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['AVTOT'] <= xhigh]
         sns.distplot(temp['AVTOT'],bins=100, kde=False)

Out[74]: <matplotlib.axes._subplots.AxesSubplot at 0x152385400>
```



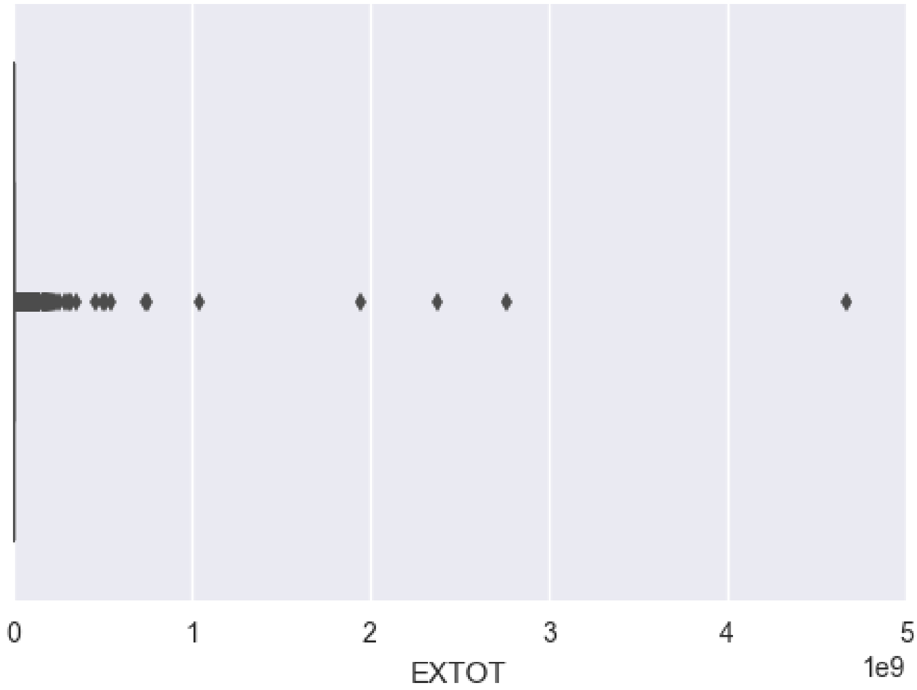
```
In [75]: mydata['EXLAND'].count() * 100 / numrecords

Out[75]: 100.0

In [76]: sns.boxplot(x='EXLAND', data=mydata)

Out[76]: <matplotlib.axes._subplots.AxesSubplot at 0x12cda45f8>
```



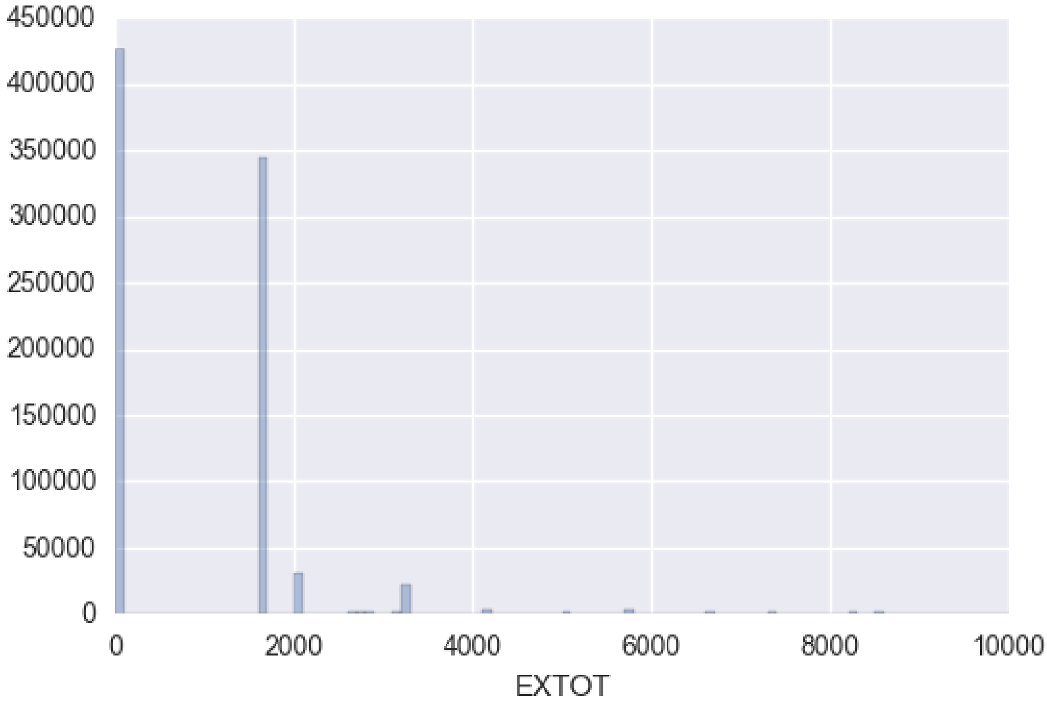
```
In [77]: xhigh = 20000
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['EXLAND'] <= xhigh]
         sns.distplot(temp['EXLAND'],bins=100, kde=False)

Out[77]: <matplotlib.axes._subplots.AxesSubplot at 0x128746eb8>
```



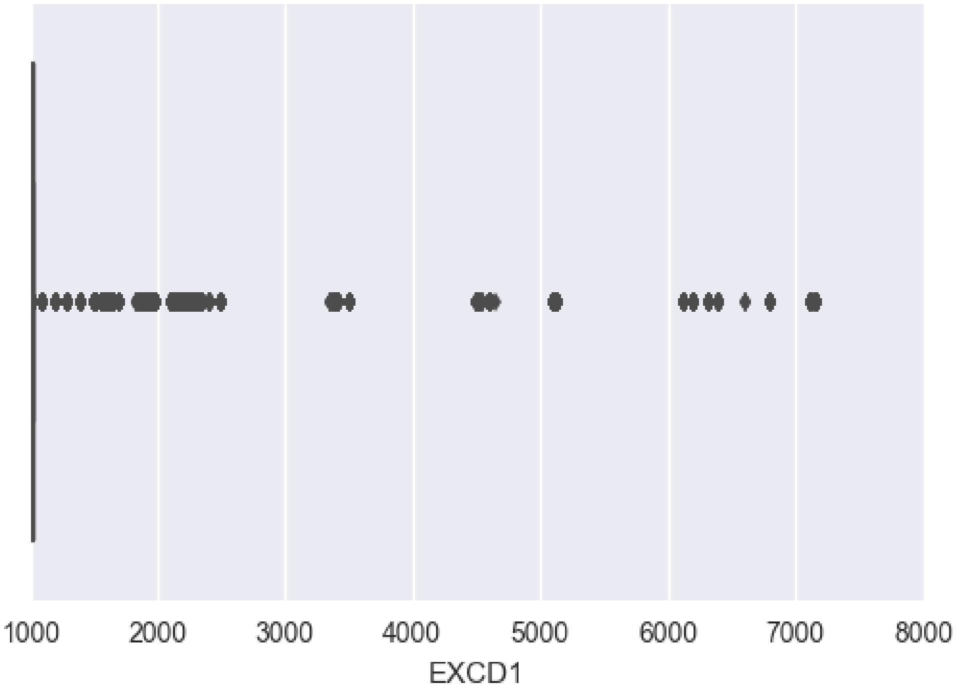
```
In [78]: mydata['EXTOT'].count() * 100 / numrecords

Out[78]: 100.0

In [79]: sns.boxplot(x='EXTOT', data=mydata)

Out[79]: <matplotlib.axes._subplots.AxesSubplot at 0x12cda95f8>
```



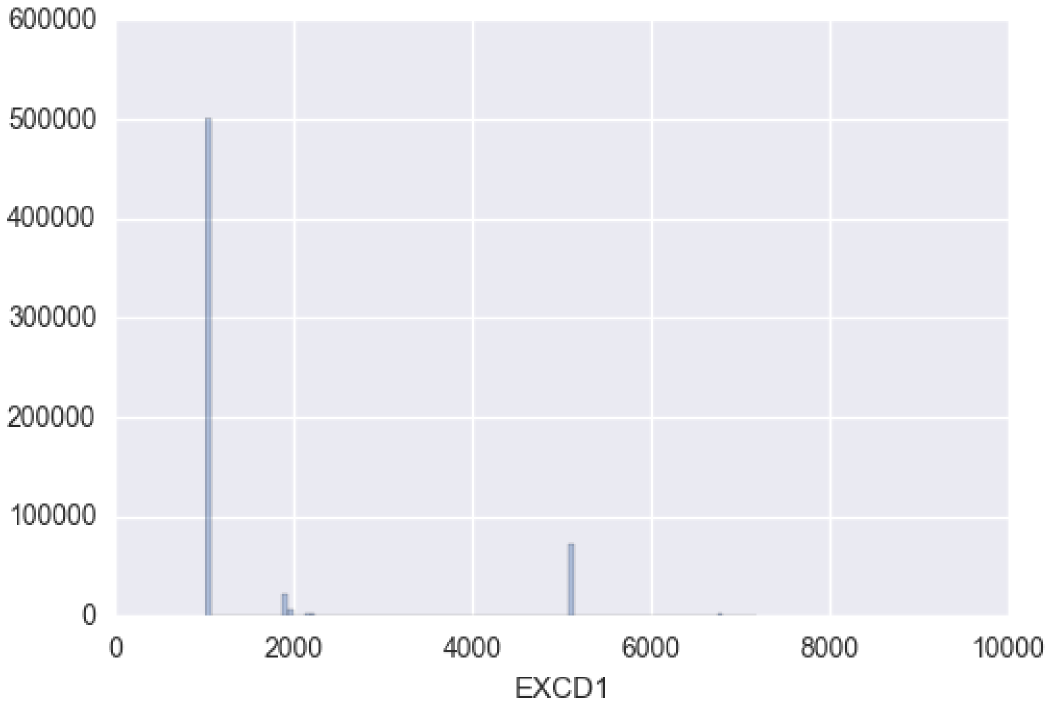
```
In [80]: xhigh = 10000
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['EXTOT'] <= xhigh]
         sns.distplot(temp['EXTOT'],bins=100, kde=False)

Out[80]: <matplotlib.axes._subplots.AxesSubplot at 0x12d0cc9b0>
```



```
In [81]: mydata['EXCD1'].count() * 100 / numrecords

Out[81]: 59.37982500059605

In [82]: sns.boxplot(x='EXCD1', data=mydata)

Out[82]: <matplotlib.axes._subplots.AxesSubplot at 0x12c7f05c0>
```



```
In [83]: xhigh = 10000
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['EXCD1'] <= xhigh]
         sns.distplot(temp['EXCD1'],bins=100, kde=False)

Out[83]: <matplotlib.axes._subplots.AxesSubplot at 0x12e0ab940>
```



```
In [84]: mydata['BLDFRONT'].count() * 100 / numrecords

Out[84]: 100.0

In [85]: mydata['STADDR'].count() * 100 / numrecords

Out[85]: 99.938869418019692

In [86]: len(mydata['STADDR'].unique())

Out[86]: 820638

In [87]: mydata['STADDR'].value_counts()

Out[87]: 501 SURF AVENUE                902
         330 EAST 38 STREET             817
         322 WEST 57 STREET             720
         155 WEST 68 STREET             671
         20 WEST 64 STREET              657
         1 IRVING PLACE                 650
         220 RIVERSIDE BOULEVARD        628
         360 FURMAN STREET              599
         200 EAST 66 STREET             585
         30 WEST 63 STREET              562
         350 WEST 42 STREET             556
```

```
2 BAY CLUB DRIVE            556
200 RECTOR PLACE            549
301 EAST 79 STREET          538
350 WEST 50 STREET          498
630 1 AVENUE                488
635 WEST 42 STREET          483
88 GREENWICH STREET         453
150 WEST 51 STREET          447
99 JOHN STREET              445
25 CENTRAL PARK WEST        441
138-35 ELDER AVENUE         437
1623 3 AVENUE               434
1 BAY CLUB DRIVE            427
5 EAST 22 STREET            426
310 WEST 52 STREET          425
106 CENTRAL PARK SOUTH      420
382 CENTRAL PARK WEST       415
400 CENTRAL PARK WEST       415
25-40 SHORE BOULEVARD       415
                           ...
1258 EVERGREEN AVENUE         1
4032 MURDOCK AVENUE           1
45-39 170 STREET              1
147-55 28 AVENUE              1
122-06 LAX AVENUE             1
92 NORTH MADA AVENUE          1
122 WEST 81 STREET            1
1829 WEST 5 STREET            1
7208 NARROWS AVENUE           1
130 DONGAN HILLS AVENUE       1
22 CLARKSON AVENUE            1
27-38 HUMPHREYS STREET        1
149 BAINBRIDGE STREET         1
446 EAST 77 STREET            1
2245 MILL AVENUE              1
165 EAST 35 STREET            1
1261 76 STREET                1
1440 METROPOLITAN AVENUE      1
88-33 214 STREET              1
146-26 181 STREET             1
1000 PENNSYLVANIA AVENUE      1
97-13 103 AVENUE              1
809 UNION STREET              1
BEACH 52 STREET               1
68-31 79 STREET               1
310 FOREST AVENUE             1
14-04 209 STREET              1
1863 CROPSEY AVENUE           1
```

```
         25 PELTON AVENUE                 1
         118-12 194 STREET                1
         Name: STADDR, dtype: int64
```

```
In [88]: mydata['ZIP'].count() * 100 / numrecords
```

```
Out[88]: 97.486493574613164
```

```
In [89]: len(mydata['ZIP'].unique())
```

```
Out[89]: 197
```

```
In [90]: mydata['ZIP'].value_counts()
```

```
Out[90]: 10314.0    24605
         11234.0    20001
         10462.0    16905
         10306.0    16576
         11236.0    15678
         11385.0    14921
         11229.0    12793
         11211.0    12710
         10312.0    12634
         11207.0    12293
         11215.0    11834
         11235.0    11312
         11203.0    11241
         11208.0    11139
         11204.0    11061
         10469.0    11030
         11214.0    10886
         11223.0    10741
         10305.0    10624
         11434.0    10505
         11355.0    10492
         11219.0    10300
         11357.0     9851
         11413.0     9784
         11373.0     9779
         11220.0     9686
         10023.0     9518
         10016.0     9362
         10019.0     9355
         10304.0     9333
                    ...
         10475.0      687
         10034.0      650
         10039.0      596
         10044.0      588
         10040.0      546
         10037.0      526
         11040.0      450
         11239.0      195
         11109.0      194
         11243.0      185
         10020.0      120
         10803.0       46
         10282.0       22
         11430.0       14
         10309.0       14
         11697.0       10
         11227.0        5
         33803.0        3
         10281.0        3
         11696.0        2
         11695.0        2
         10307.0        2
         11242.0        2
         10048.0        2
         11241.0        1
         11371.0        1
         11005.0        1
         11359.0        1
         11352.0        1
         10162.0        1
         Name: ZIP, dtype: int64
```

```
In [91]: mydata['EXMPTCL'].count() * 100 / numrecords
```

```
Out[91]: 1.4297498986720072
```

```
In [92]: len(mydata['EXMPTCL'].unique())
```

```
Out[92]: 15
```

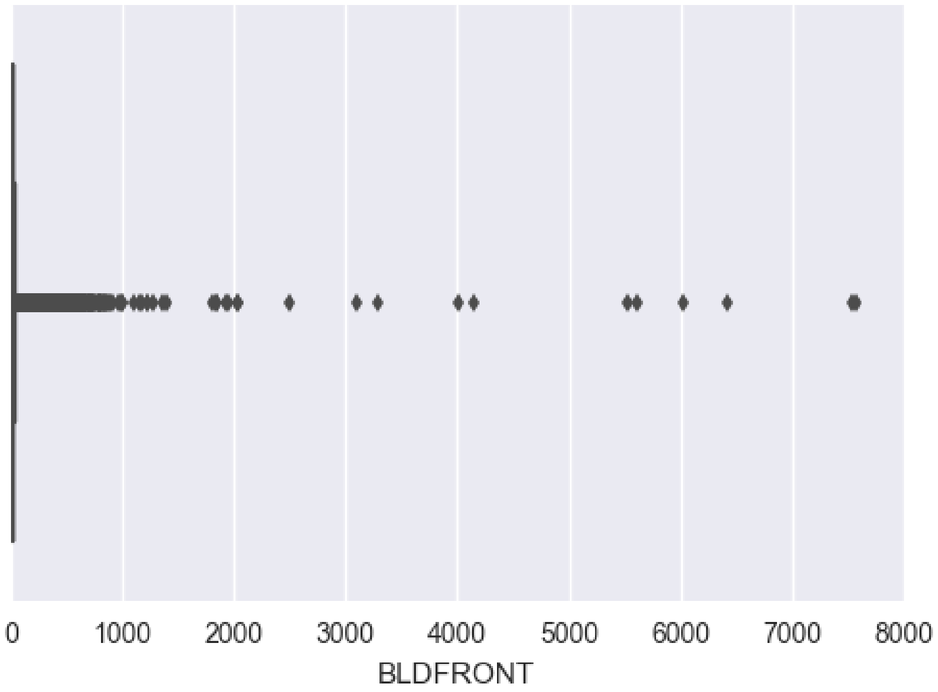
```
In [93]: mydata['EXMPTCL'].value_counts()
```

```
Out[93]: X1    6494
         X5    5158
         X7     818
         X6     760
         X2     665
         X4     438
         X8     289
         X3     260
         X9     105
         VI       1
         KI       1
         R4        1
         5         1
         A9        1
         Name: EXMPTCL, dtype: int64
```

```
In [ ]:
```

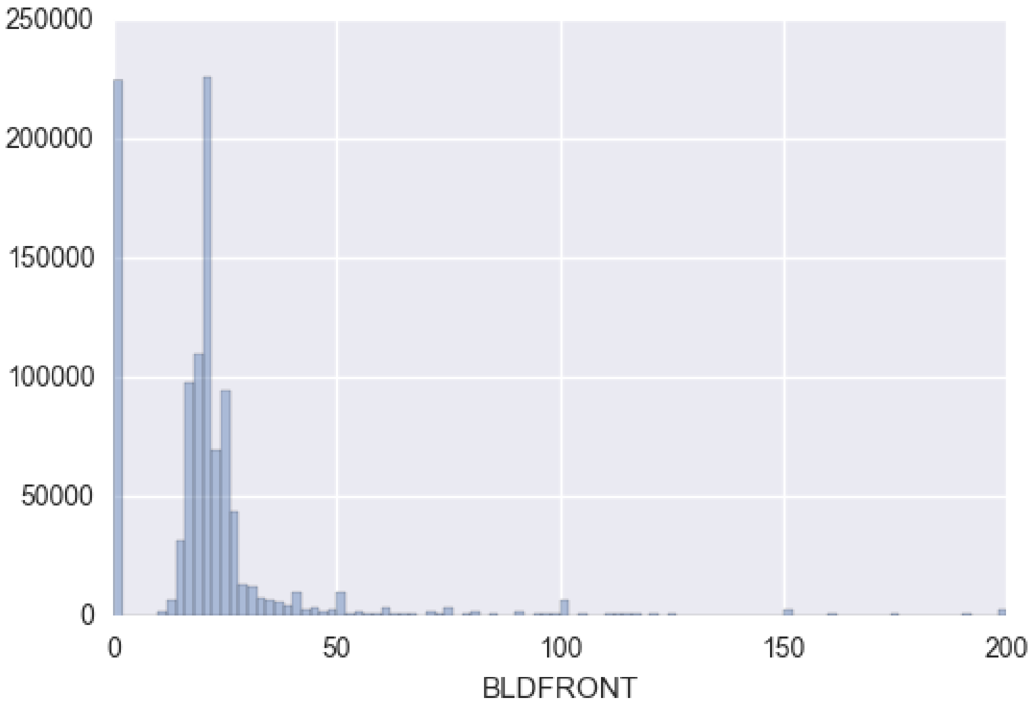
```
In [94]: sns.boxplot(x='BLDFRONT', data=mydata)
```

```
Out[94]: <matplotlib.axes._subplots.AxesSubplot at 0x12d417860>
```

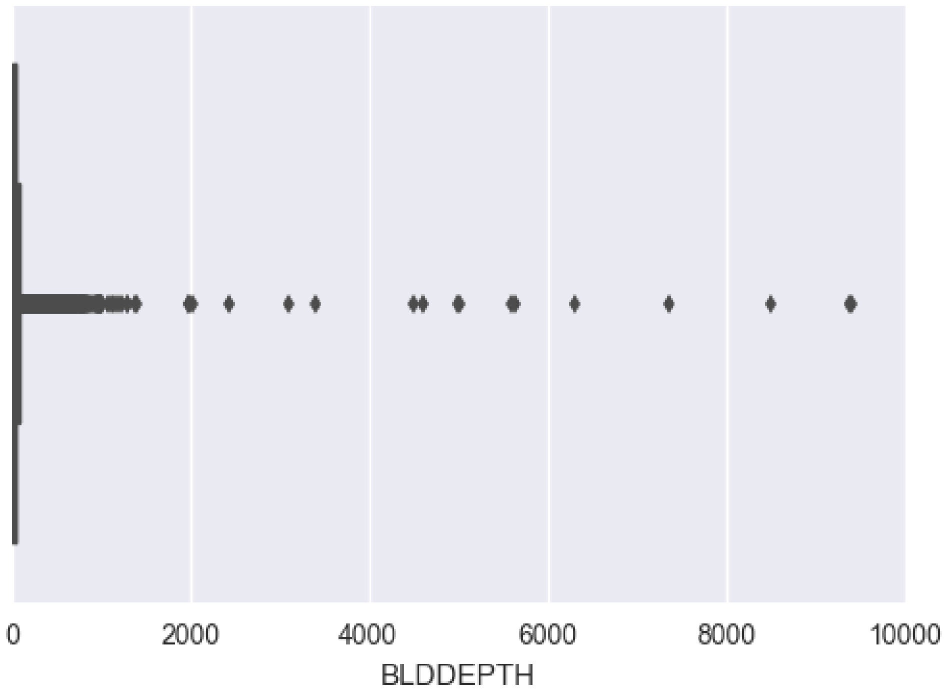


```
In [95]: xhigh = 200
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['BLDFRONT'] <= xhigh]
         sns.distplot(temp['BLDFRONT'],bins=100, kde=False)
```

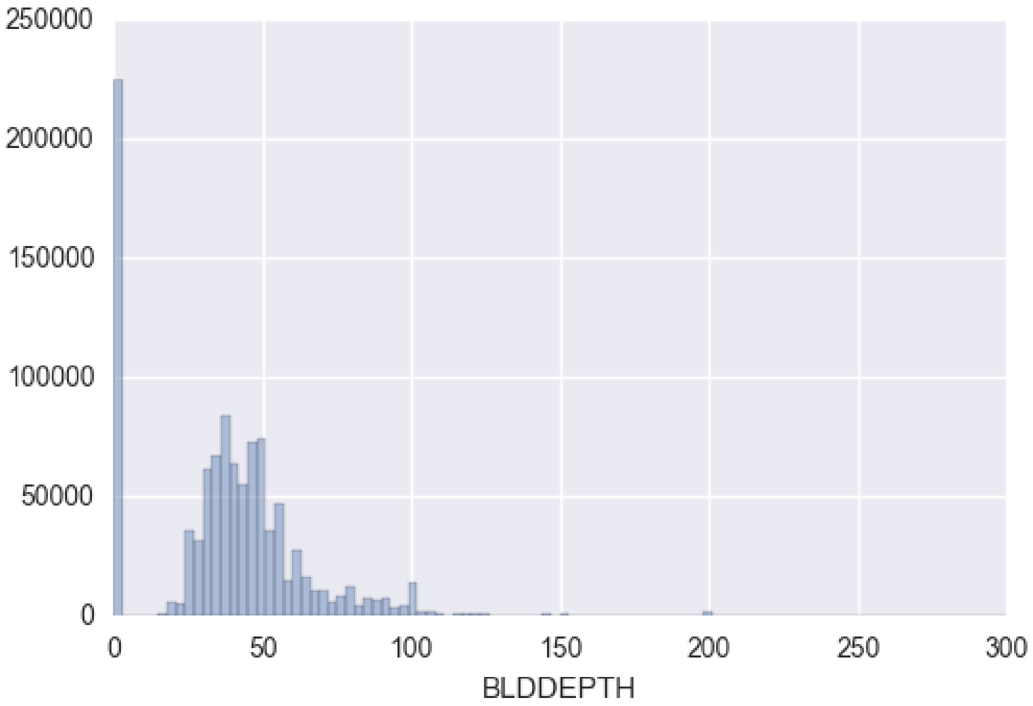
```
Out[95]: <matplotlib.axes._subplots.AxesSubplot at 0x12aa2bdd8>
```



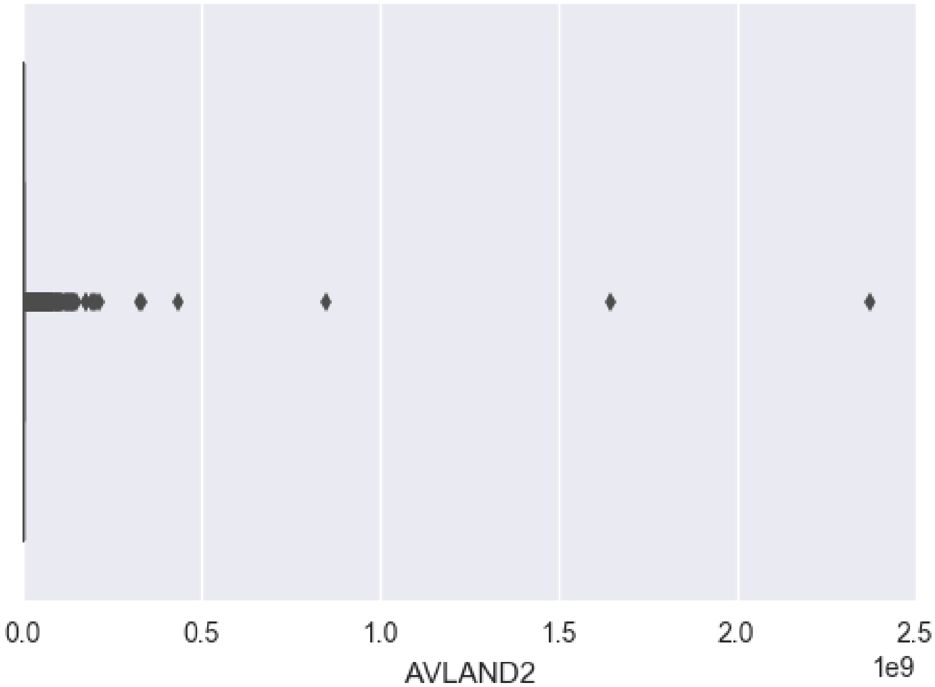
```
In [96]: mydata['BLDDEPTH'].count() * 100 / numrecords

Out[96]: 100.0

In [97]: sns.boxplot(x='BLDDEPTH', data=mydata)

Out[97]: <matplotlib.axes._subplots.AxesSubplot at 0x12a7fe470>
```



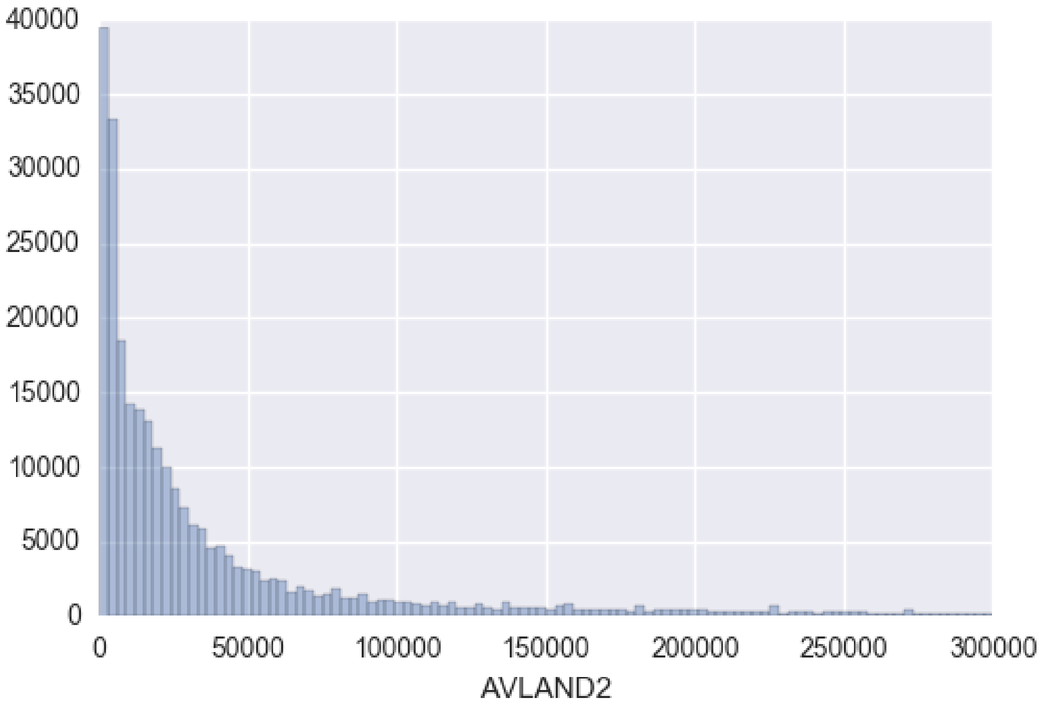
```
In [98]: xhigh = 300
         sns.plt.xlim(0,xhigh)
         temp = mydata[mydata['BLDDEPTH'] <= xhigh]
         sns.distplot(temp['BLDDEPTH'],bins=100, kde=False)

Out[98]: <matplotlib.axes._subplots.AxesSubplot at 0x11c924c50>
```



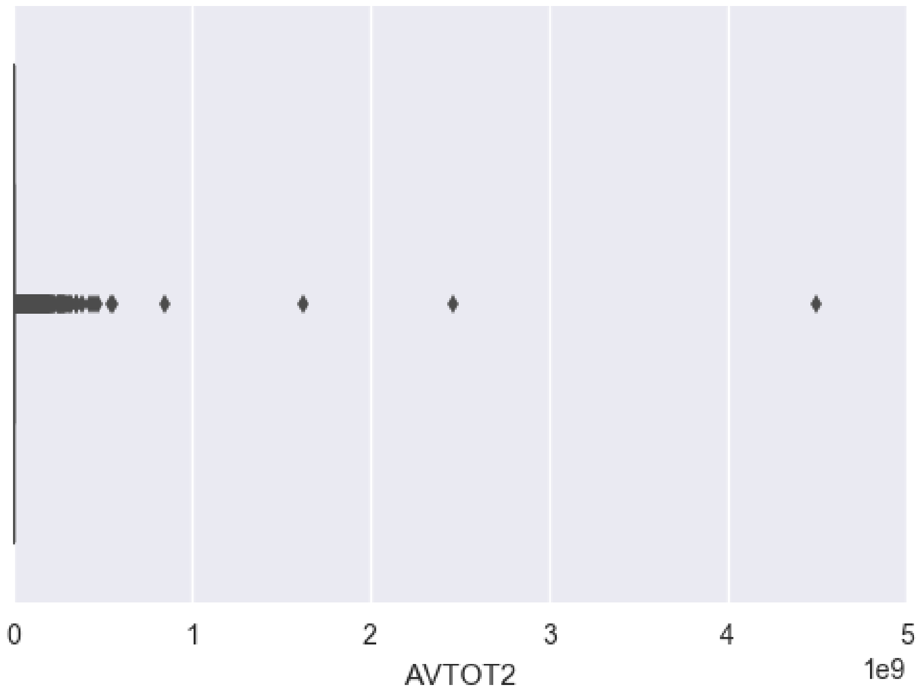
```
In [99]: mydata['AVLAND2'].count() * 100 / numrecords

Out[99]: 26.795031352073053

In [100]: mydata['AVLAND2'].count() * 100/ numrecords

Out[100]: 26.795031352073053

In [101]: sns.boxplot(x='AVLAND2', data=mydata)

Out[101]: <matplotlib.axes._subplots.AxesSubplot at 0x11c8bf898>
```



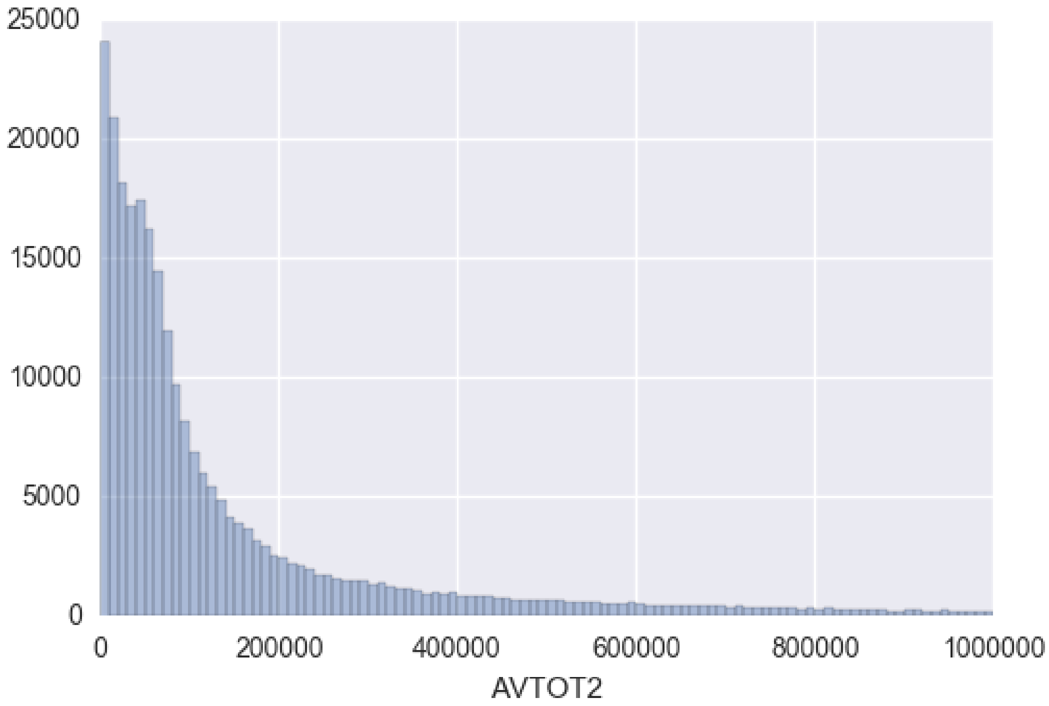
```
In [102]: xhigh = 300000
          sns.plt.xlim(0,xhigh)
          temp = mydata[mydata['AVLAND2'] <= xhigh]
          sns.distplot(temp['AVLAND2'],bins=100, kde=False)

Out[102]: <matplotlib.axes._subplots.AxesSubplot at 0x1289e2320>
```



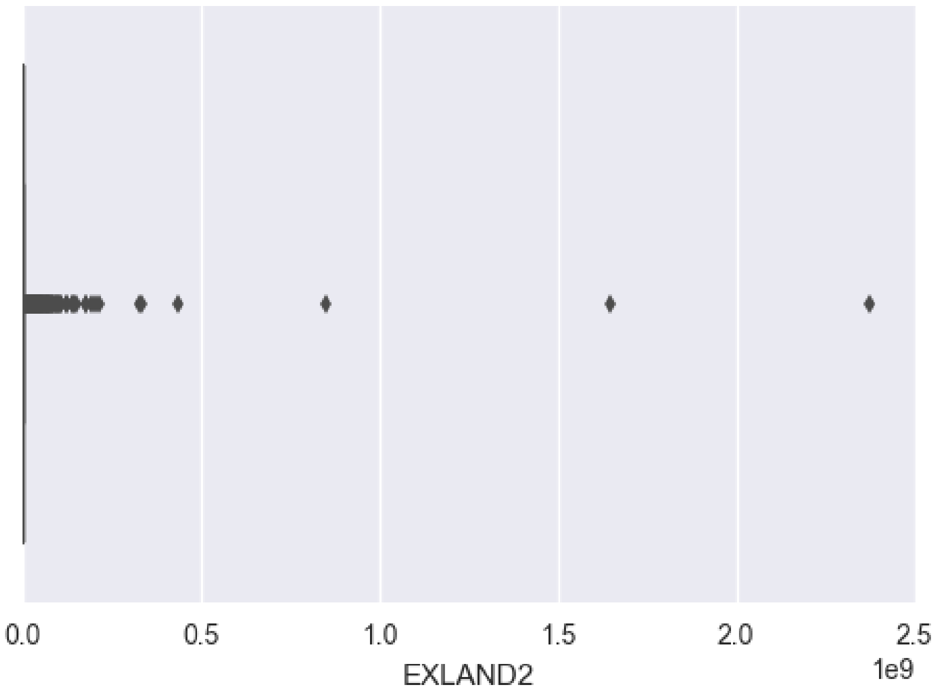
```
In [103]: mydata['AVTOT2'].count() * 100 / numrecords

Out[103]: 26.795603557208594

In [104]: sns.boxplot(x='AVTOT2', data=mydata)

Out[104]: <matplotlib.axes._subplots.AxesSubplot at 0x12db33780>
```



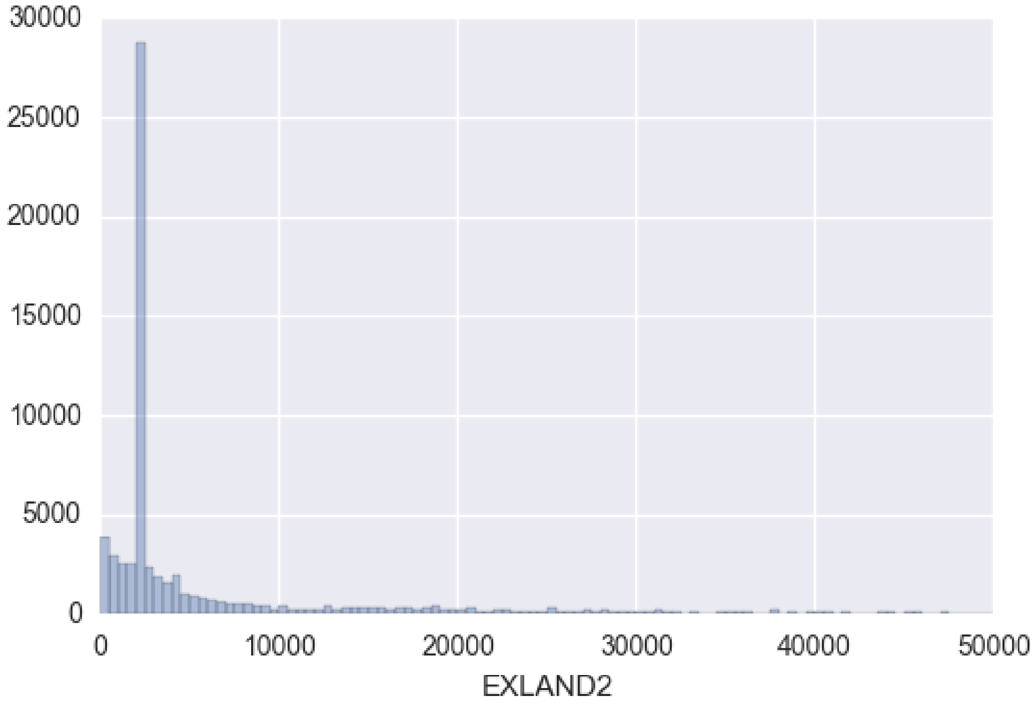
```
In [105]: xhigh = 1000000
          sns.plt.xlim(0,xhigh)
          temp = mydata[mydata['AVTOT2'] <= xhigh]
          sns.distplot(temp['AVTOT2'],bins=100, kde=False)

Out[105]: <matplotlib.axes._subplots.AxesSubplot at 0x11d8f5668>
```



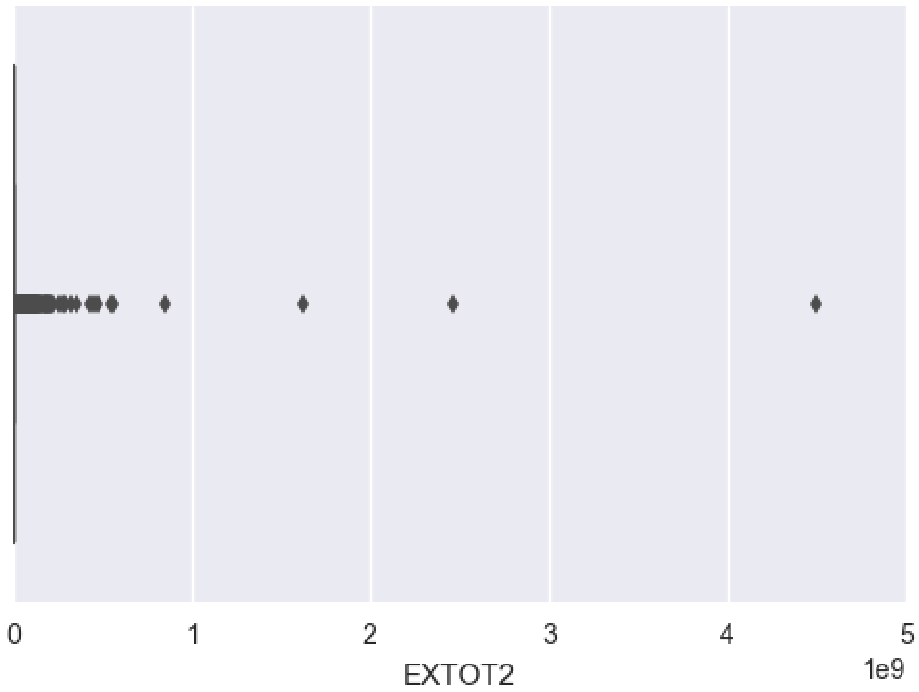
```
In [106]: mydata['EXLAND2'].count() * 100 / numrecords

Out[106]: 8.265980020504017

In [107]: sns.boxplot(x='EXLAND2', data =mydata)

Out[107]: <matplotlib.axes._subplots.AxesSubplot at 0x1236e8908>
```



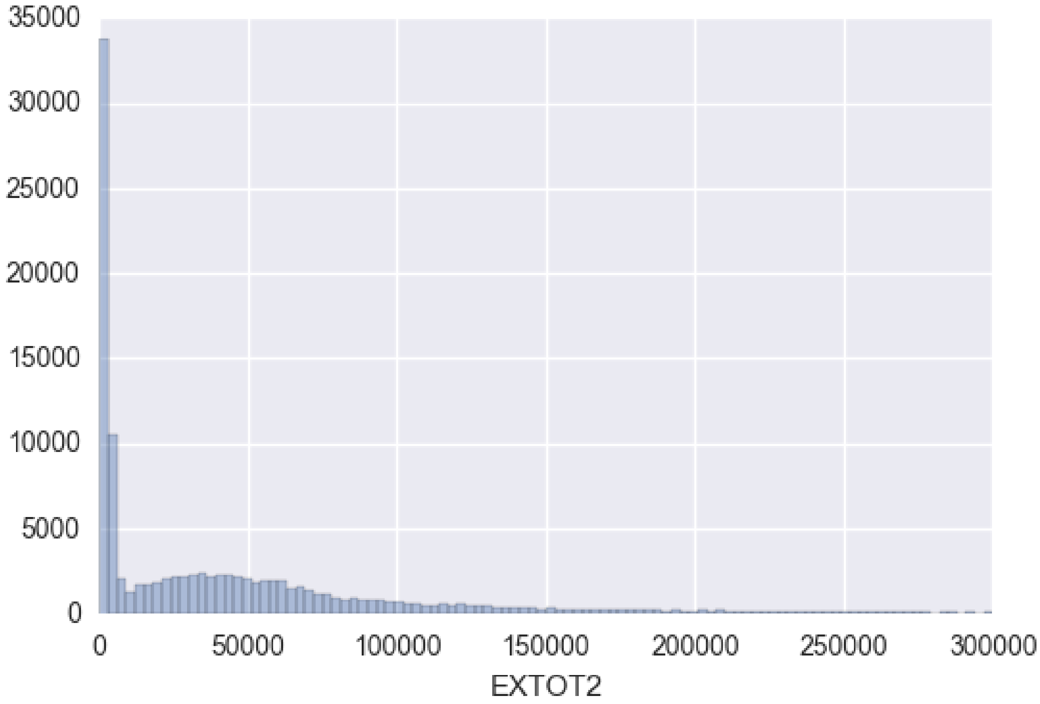
```
In [108]: xhigh = 50000
          sns.plt.xlim(0,xhigh)
          temp = mydata[mydata['EXLAND2'] <= xhigh]
          sns.distplot(temp['EXLAND2'],bins=100, kde=False)

Out[108]: <matplotlib.axes._subplots.AxesSubplot at 0x1255a97f0>
```



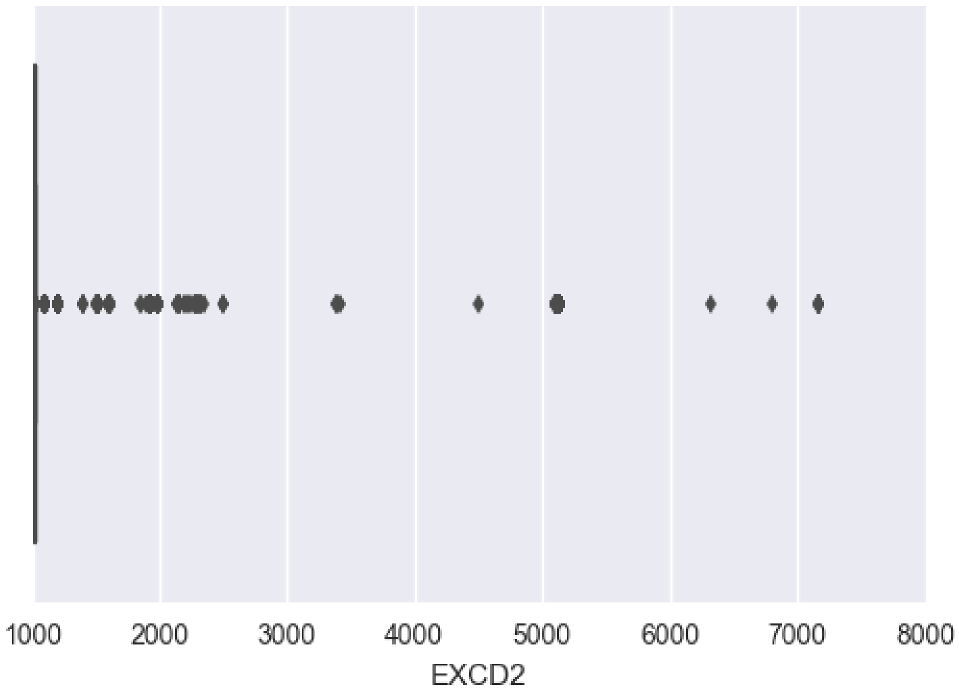
```
In [109]: mydata['EXTOT2'].count() * 100 / numrecords

Out[109]: 12.391388312710106

In [110]: sns.boxplot(x='EXTOT2', data=mydata)

Out[110]: <matplotlib.axes._subplots.AxesSubplot at 0x129f97fd0>
```



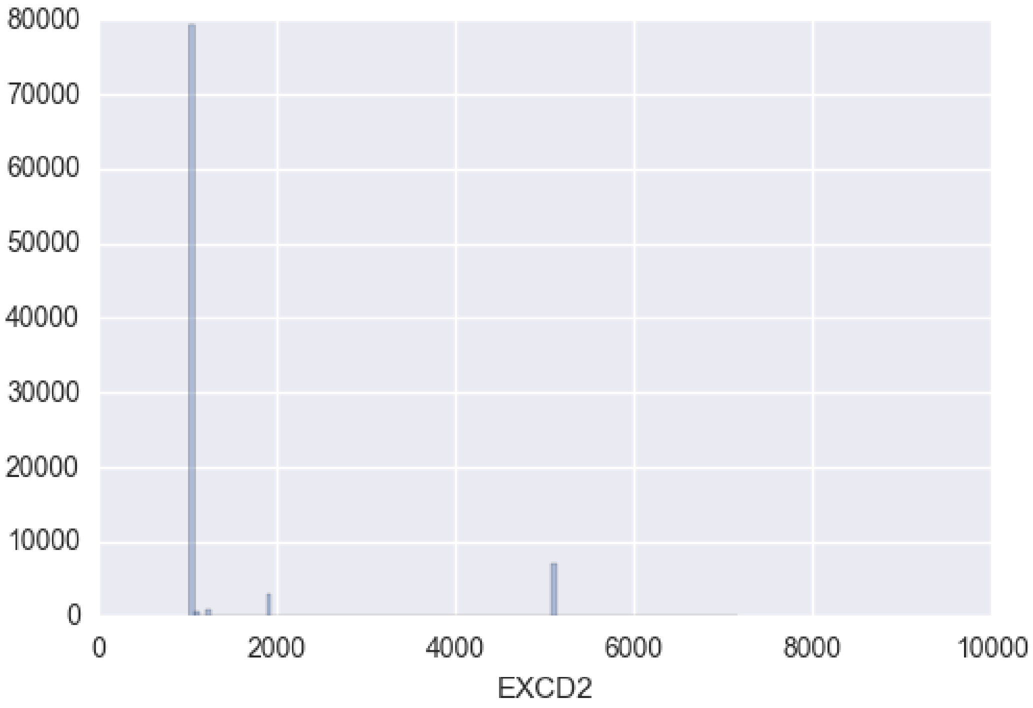
```
In [111]: xhigh = 300000
          sns.plt.xlim(0,xhigh)
          temp = mydata[mydata['EXTOT2'] <= xhigh]
          sns.distplot(temp['EXTOT2'],bins=100, kde=False)

Out[111]: <matplotlib.axes._subplots.AxesSubplot at 0x12d914898>
```



```
In [112]: mydata['EXCD2'].count() * 100 / numrecords

Out[112]: 8.6728178718737325

In [113]: sns.boxplot(x='EXCD2', data=mydata)

Out[113]: <matplotlib.axes._subplots.AxesSubplot at 0x1274b97b8>
```

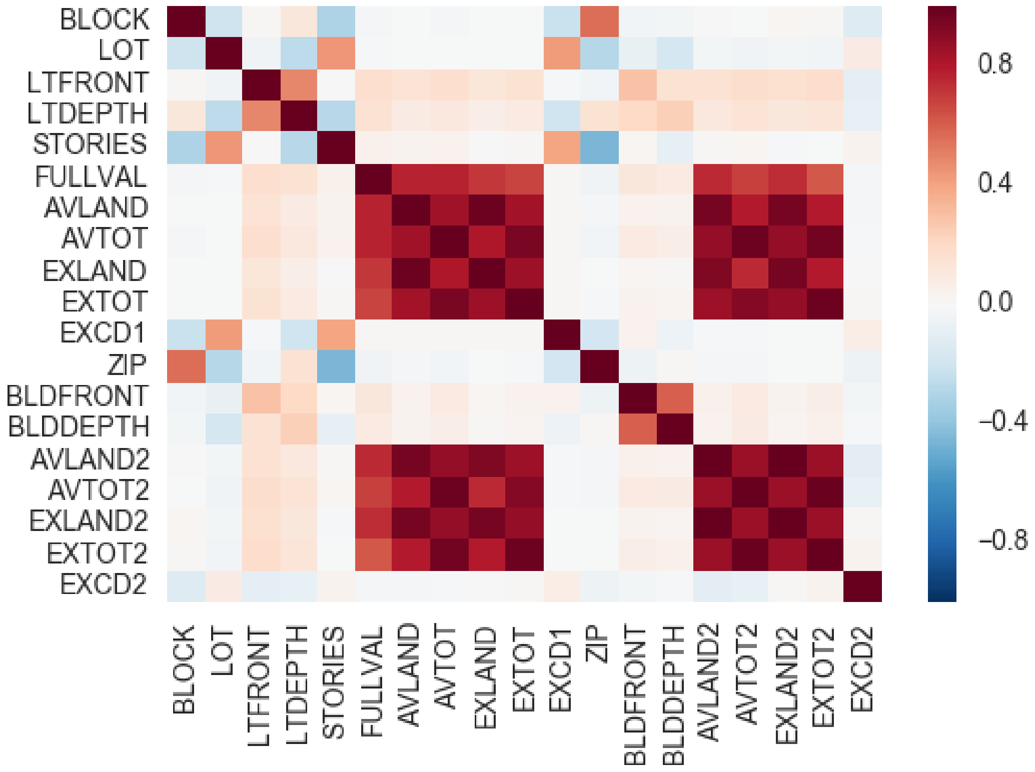


```
In [114]: xhigh = 10000
          sns.plt.xlim(0,xhigh)
          temp = mydata[mydata['EXCD2'] <= xhigh]
          sns.distplot(temp['EXCD2'],bins=100, kde=False)

Out[114]: <matplotlib.axes._subplots.AxesSubplot at 0x11dda5d68>
```



```
In [115]: mydata['PERIOD'].count() * 100 / numrecords

Out[115]: 100.0

In [116]: len(mydata['PERIOD'].unique())

Out[116]: 1

In [117]: mydata['PERIOD'].value_counts()

Out[117]: FINAL    1048575
          Name: PERIOD, dtype: int64

In [118]: mydata['YEAR'].count() * 100 / numrecords

Out[118]: 100.0

In [119]: len(mydata['YEAR'].unique())

Out[119]: 1

In [120]: mydata['YEAR'].value_counts()

Out[120]: 2010/11    1048575
          Name: YEAR, dtype: int64
```

```
In [121]: mydata['VALTYPE'].count() * 100 / numrecords

Out[121]: 100.0

In [122]: len(mydata['VALTYPE'].unique())

Out[122]: 1

In [123]: mydata['VALTYPE'].value_counts()

Out[123]: AC-TR    1048575
          Name: VALTYPE, dtype: int64

In [124]: sns.heatmap(mydata.corr())

Out[124]: <matplotlib.axes._subplots.AxesSubplot at 0x12b6c9320>
```



```
In [ ]:
```